NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A MULTI-ARMED BANDIT APPROACH TO SUPERQUANTILE SELECTION

by

Adam J. Hepworth

June 2017

Thesis Advisor:	Roberto S. Szechtman
Second Reader:	Michael P. Atkinson

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2017	Master's Thesis 10-31-2016 to 06-16-2017

4. TITLE AND SUBTITLE

A MULTI-ARMED BANDIT APPROACH TO SUPERQUANTILE SELECTION

5. FUNDING NUMBERS

6. AUTHOR(S)

Adam J. Hepworth

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

We study a resource allocation problem in an intelligence setting. The intelligence cycle is comprised of three phases: collection, processing, and analysis. Enhanced efficiency within the first two stages directly impacts the number and types of important items that are considered by analysts, increasing the frequency of the most important documents that are reviewed. The dilemma here is that an analyst needs to quickly determine which sources to investigate, in order to provide meaningful analysis to a request for information with, a concrete deadline. Initially, the value of each source is unknown; so, too, is the probabilistic nature of the value derived from each item. Generally, more sources and documents are available to be considered within a limited time frame than could be ever analyzed, compounding the complexity of this problem. Our goal is to efficiently find the source that produces the largest fraction of relevant items with respect to a request for information. By "efficiently," we mean that the analyst balances exploration versus exploitation of the different sources judiciously. As such, the theoretical framework for this problem is that of a multi-armed bandit, a classic iterative decision learning process. This thesis presents a new approach to identifying the optimal arm(s) of a multi-arm bandit with the largest or smallest quantile or superquantile risk, under a loss constraint. This problem is not only important in intelligence applications, but in marketing and finance. We extend the existing theoretical framework of dealing with quantiles to a novel situation with estimators of conditional expectations over an unknown quantile. Two sequential elimination algorithms are developed that select the most important source for a given constraint level, sampling from the arm(s) with the largest conditional expectation over a quantile.

14. SUBJECT TERMS

quantile, superquantile, multi-armed bandit, value-at-risk, conditional value-at-risk, probably approximately correct, root finding, stochastic root finding, loss constraint, applied probability theory, iterative decision learning, machine learning, intelligence processing, intelligence cycle, quantitative finance.

15. NUMBER OF PAGES 73

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

A MULTI-ARMED BANDIT APPROACH TO SUPERQUANTILE SELECTION

Adam J. Hepworth
Captain, Australian Army
B.Sc., The University of New South Wales, December 2009
M.Log&SupChMgmt, The University of South Australia, August 2015

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2017

Approved by: Roberto S. Szechtman
Thesis Advisor

Michael P. Atkinson
Second Reader

Patricia A. Jacobs
Chair, Operations Research Department

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

We study a resource allocation problem in an intelligence setting. The intelligence cycle is comprised of three phases: collection, processing, and analysis. Enhanced efficiency within the first two stages directly impacts the number and types of important items that are considered by analysts, increasing the frequency of the most important documents that are reviewed. The dilemma here is that an analyst needs to quickly determine which sources to investigate in order to provide meaningful analysis to a request for information with a concrete deadline. Initially, the value of each source is unknown; so, too, is the probabilistic nature of the value derived from each item. Generally, more sources and documents are available to be considered within a limited time frame than could be ever analyzed, compounding the complexity of this problem. Our goal is to efficiently find the source that produces the largest fraction of relevant items with respect to a request for information. By "efficiently," we mean that the analyst balances exploration versus exploitation of the different sources judiciously. As such, the theoretical framework for this problem is that of a multi-armed bandit, a classic iterative decision learning process. This thesis presents a new approach to identifying the optimal arm(s) of a multi-arm bandit with the largest or smallest quantile or superquantile risk, under a loss constraint. This problem is not only important in intelligence applications, but in marketing and finance. We extend the existing theoretical framework of dealing with quantiles to a novel situation with estimators of conditional expectations over an unknown quantile. Two sequential elimination algorithms are developed that select the most important source for a given constraint level, sampling from the arm(s) with the largest conditional expectation over a quantile.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

CDF Cumulative Distribution Function

CVaR Conditional Value-at-Risk

DTMC Discrete Time Markov Chain

JP Joint Publication

MAB Multi-armed Bandit

NPS Naval Postgraduate School

OR Operations Research

PAC Probably Approximately Correct

pdf Probability Density Function

RV Random Variable

SAA Sample Average Approximation

QPAC Qualitative Probably Approximately Correct

QUCB Qualitative Upper Confidence Bound

UCB Upper Confidence Bound

VaR Value-at-Risk

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

The intelligence cycle can be considered to consist of three broad phases: collection, processing, and analysis. As part of the second phase, the goal is to pass items that are important, in relation to a request for information, for processing by senior analysts. The objective is to create efficiencies within the processing phase, leading to a reduction in required analysts for a task, and better utilizing the total analyst resource. The dilemma here is that an analyst needs to quickly determine which sources to investigate, in order to provide meaningful analysis to a request for information with a finite and concrete deadline. To add further complexity, there generally exists more sources and documents than can ever be analyzed within the time frame given. Our goal in this thesis is to produce algorithms to efficiently discover the most relevant intelligence source(s) to analyze in order to have analysts spend less time processing data and more time to deliver critical insights. The essence of this work is a resource allocation problem within an intelligence setting and we derive the following organizational impacts as our primary motivation:

1. a decrease in the total time that an analyst spends processing data,
2. a decrease in the required number of analysts for a particular task, resulting in a reduction of resource allocation waste,
3. an increase in the time that each analyst delivers insights from their analysis of intelligence information,
4. an increase in the total intelligence product output of an agency or organization, and
5. an increase in the tempo of an agency or organization: delivering more intelligence faster.

Our theoretical framework for this problem is that of a stochastic multi-armed bandit, a classic iterative probabilistic decision learning problem. The goal of the multi-armed bandit is to determine the optimal trade-off between exploration and exploitation. The classic multi-armed bandit problem concept stems from observing gamblers within a casino playing slot machines—the term bandit stemming from the colloquial gambler term for a slot machine—the gambler must choose the number of times to play each machine as well as the order to play them. When a bandit is pulled, an immediate random reward is observed from an underlying probability distribution specific to that individual machine, and unknown to the gambler. The gambler's objective in the game is to the maximize the cumulative reward over the number of plays. Within the stochastic setting, we note that for this problem each arm of the bandit can have a distinct probability distribution that determines the sequence of the rewards observed.

In this thesis, we address a new approach to identifying the optimal arm(s) of a bandit with the largest or smallest quantile or superquantile risk, under constraints. This is analogous to a root

finding problem in a stochastic setting and is not only important in intelligence applications, but also within on-line marketing and quantitative finance. Quantile risk, more commonly known as value-at-risk, is one of the most systemic risk metrics within the financial engineering community. The superquantile risk is an improved metric known within quantitative finance community as conditional value-at-risk and is a coherent [1], regular [2], and convex [3] measure that seeks to model the distributional behaviour of risk, quantifying expected losses that may be seen within the tail [4].

We extend the existing theoretical framework of dealing with quantiles as seen in [5] and [6], to a novel situation with estimators of conditional expectations over an unknown quantile. Two sequential elimination algorithms are developed that select the most important source for a given constraint level, sampling from the arm(s) with the largest conditional expectation over a quantile. In the aforementioned intelligence setting, this translates into efficiently determining the source that produces the largest fraction of items of a given quality on average; the idea being that each request for information has a particular quality stipulation.

References

[1] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, pp. 201–227, 1999.

[2] R. Rockafellar and S. Uryasev, "The fundamental risk quadrangle in risk management, optimization and statistical estimation," *Surveys in Operations Research and Management Science*, vol. 18, pp. 33–53, 2013.

[3] A. Ruszczynski and A. Shapiro, "Optimization of convex risk functions," *Mathematics of Operations Research*, vol. 31(3), pp. 433–452, 2006.

[4] R. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking and Finance*, vol. 26, pp. 1443–1471, 2002.

[5] B. Szörényi, R. Busa-Fekete, P. Weng, and E. Hüllermeier, "Qualitative multi-armed bandits: A quantile-based approach," *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015.

[6] P. Glynn and S. Juneja, "Ordinal optimization - empirical large deviations rate estimators, and stochastic multi-armed bandits," *arXiv:1507.04564*, 2015.

Acknowledgments

Without the many hours of thought-provoking reading, insightful discussion and laughter with Roberto Szechtman, this research would be little more than a fleeting idea. I've gained vast insights from the vast topics, books and papers we discussed in the nine months leading up to the official beginning of my thesis topic: merely the start of this journey. I often pondered if Roberto thought I was just participating in a personalised seminar lecture series with him, as opposed to ever undertaking any real research! Thankfully, after many (many, many...) iterations of defining a problem, we had, as Roberto would say, a *"well-defined, non-trivial problem"*: he wasn't kidding! And, whilst not the most important element on a thank-you list, Roberto's consistent need for coffee and a refusal to let me ever buy for him was a perfect match! Roberto, thank you for your inspiring mentorship, and what I hope will become a lifelong friendship. I could not have asked for a better advisor.

Mike Atkinson went well beyond what I could have ever asked in the role as a second reader, being an integral part of the team and helping to navigate through the dense mathematics: a sounding board and advisor in his own right. Mike, your assistance in co-authoring a paper with Roberto and me was fantastic and I will be forever grateful. Thank you for providing me with the foundational knowledge in stochastic modelling, as without this, I would have never taken up this thesis topic.

In thanking the wider Operations Research Department community here at the Naval Postgraduate School, I'd like to thank Matt Carlyle, Tom Lucas, Connor McLemore, and Jeff Kline. Your guidance offered throughout the program has been invaluable to my development. I'm very thankful that the faculty of Applied Mathematics Department (particularly Carlos Borges) allowed me to keep coming back for my second program, albeit with a joviality: it's been great! I'd like to make special mention of Jeff House for the many hours of intellectual conversation on a myriad of academic, life and philosophical topics. Thank you for the lesson that you'll never know where you'll be when the world changes; a humbling story that I'll never forget.

At Naval Postgraduate School, I've made a number of close and lifelong friends—too many to name here—you know who you are: thank you for the unforgettable time in Monterey. Whilst living in California I've had the honour of hosting many close friends from Australia and providing them with an insight into the life within the Operations Research program here. Their interest and enthusiasm have been fantastic, and I thank them for enjoying my passion with me. Hearing a familiar voice during my time living in California was always a delight.

In closing, I'd like to thank those closest to me. Firstly, dearest Molly, thanks for being you. Your

love and support over the past two years have been awesome, just like you; I'll see you soon! Lastly, but certainly not least, my parents. Their unwavering support of all that I do is amazing; their pride in me knows no bounds. I am forever grateful for their eternal wisdom and foresight to prioritise education as a critical foundational aspect throughout my life, as the difference this has made to me cannot ever truly be quantified.

CHAPTER 1:
Introduction

1.1 Introduction

The purpose of this chapter is to provide an overview of the thesis, affording readers the opportunity to gain a contextual understanding. The thesis scope is given and we subsequently frame the problem by defining the operational motivation through two lenses, firstly with a direct military, and secondly a non-military application in marketing. We undertake a brief discussion of risk within the framework motivation presented, leading to the development of the research questions that follow. Chapter 1 is finalized by providing a snapshot of our contributions and the structure of the thesis that follows.

1.2 Scope

This thesis deals with intelligence analysis techniques and procedures in environments that change in real time. From the technical standpoint, we employ ideas from the machine learning and stochastic optimization communities of operations research. We develop a model, analyze, and numerically simulate its performance against constructed data. Decision support tools and performance metrics with live data is beyond the scope of this thesis, but can be easily implemented with the algorithms that appear in Chapter 3.

1.3 Motivation

Two primary settings have been considered as motivations for this research. The first application stems from intelligence operations and the second from the field of financial engineering risk management.

1.3.1 Models of Intelligence Operations

The intelligence cycle consists of five broad phases that link the direction of objectives, through collection, processing, and analysis, to outcomes for dissemination. Throughout the work presented here, we consider three key stages of information transformation, consisting of collection, processing and analysis, as shown in Figure 1.1. As such, the intelligence process can be thought of as consisting of two stages prior to intelligence dissemination and integration [1]. This perspective

presented is juxtaposed to the intelligence cycle shown in Figure 1.1 briefly. The sub-processes of the three initial stages include:

1. Collection: Raw intelligence items that are collected from a source and collated at an intelligence cell.
2. Processing: The activities are undertaken from processors to manipulate raw intelligence items and identify those that may be suitable for analysts to invest further effort in, deriving meaningful outcomes from.
3. Analysis: The professional analyst evaluates the importance of this information, and delivers output product in a timely manner that provides a warfighting advantage and tangible outcomes.

Through creating efficiencies within the first two stages of the intelligence process, we observe an improvement in the analysis phase where the majority of resourced effort is expended. This results in the most important items under consideration for a greater amount of time by the analysts. We can think of this process in terms of a signal processing analogy, where valuable intelligence can be considered the true signal and non-valuable intelligence the signal noise. Here, we attempt to remove as much of the noise from the signal as possible, whilst maximizing the time spent analyzing the true signal. The question for us is *which source should an analyst explore in any given time period?*, where the goal is *to determine which intelligence source to sample from that yield the greatest value.* The basic idea for the workflow of an intelligence request is summarized as seen in Figure 1.1 and Figure 1.2; our modelling of this problem framework consists of the following key stages:

1. A requirement for specific intelligence is received with a finite deadline by an intelligence organization.
2. A manager provides an analyst with the desired average importance level for an item, in accordance with organisational priorities.
3. The analyst now decides which source to explore, and determines the generated item importance in relation to the request for information.
4. The item is passed on if its importance is over the threshold. A source is selected so the average importance of items with importance over the threshold equals the desired value of step 2.
5. There generally exist more sources than can be feasibly explored in a given time frame, in order to obtain a relevant intelligence picture. The problem is that the analyst does not initially know which sources tend to produce a large fraction of items with importances over the threshold.
6. Exploration vs. Exploitation. As the analyst conducts an assessment of those sources, an

understanding of the source(s) that tend to yield items over the importance threshold will be attained. From here, the analyst can focus on the most reliable source(s) to deliver important items, in relation to the request for information.

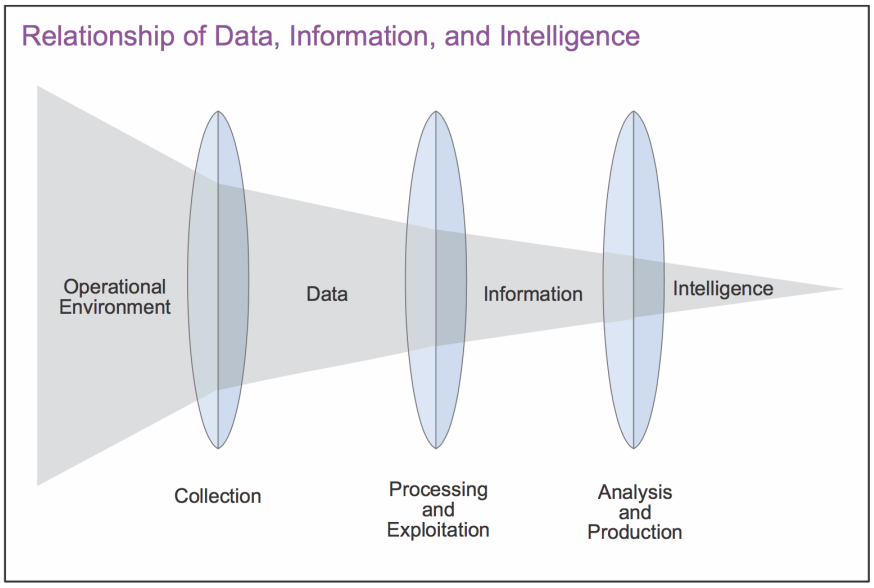


Figure 1.1. The Transformation of Data From Information Through to Intelligence. Source: [1].

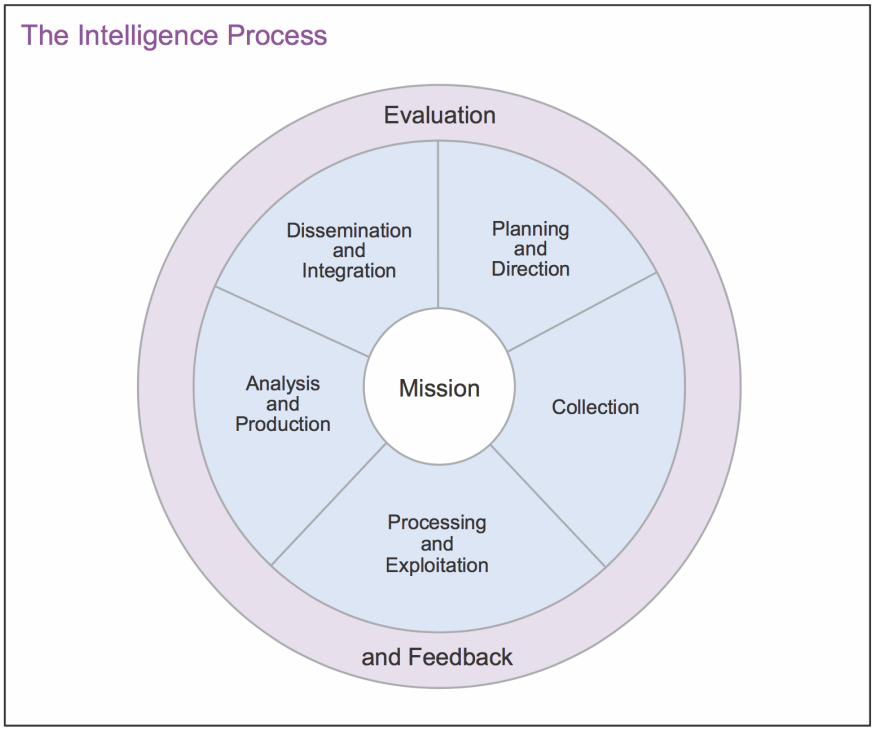


Figure 1.2. The Cycle of Intelligence Production. Source: [1].

1.4 Quantifying Risk

1.4.1 Quantile and Superquantile Risk

Quantile risk, more commonly known as Value-at-Risk (VaR), is a prevalent risk metric within the financial engineering community. The superquantile risk is an improved measure of risk, known within quantitative finance community as Conditional Value-at-Risk (CVaR), that has superior mathematical properties—see [2], [3], and [4]—that seeks to model the behaviour of risk by quantifying losses that may be seen for extreme cases [5]. In practice, each risk measure has both advantages and disadvantages; some of these are depicted in Table 1.1. A more technical discussion is provided in Chapter 2.

Table 1.1. Basic Comparison of VaR and CVaR.

Case	Value-at-Risk	Conditional Value-at-Risk
Less restrictive at the same confidence level	X	
Useful when model tails are available		X
Useful when model tails are not available	X	
Simple to optimize		X
Has mathematically superior properties		X
Risk adverse (conservative estimates)		X

This table indicates the usage of value-at-risk and conditional value-at-risk for various cases. Source: [6].

1.4.2 Risk Management and Marketing

For situations where the maximum expected loss over a threshold is given, the agent in our scenario wishes to discover the bandit arm with the largest or smallest threshold that satisfies the desired level—the loss constraint in our technical problem—or the arm with the largest or smallest probability of exceeding the threshold that meets our constraint. These problems are natural in risk portfolio analysis, where the loss threshold is known as VaR, and the expected conditional loss over the worst 100α percent scenarios is known as the CVaR at level α; for more information see [7], [3], and [8].

From an online marketing perspective, each arm corresponds to a marketing campaign for some product. The input is a number C that represents the average quality (e.g., a function of age, income, gender, etc.) of the individuals desired by the seller. The conditional value-at-risk is the fraction of people generated by the marketing campaign who have an average quality C and hence, the retailer

wishes to find the marketing campaign with the largest conditional value-at-risk. The quality of individuals generated by each marketing campaign is analogous to items generated by intelligence source.

1.5 Research Questions

We use the following guiding questions as a framework to navigate and unpack the body of work undertaken.

1. Given k systems with an unknown distribution, we seek to find the system with the largest or smallest CVaR or VaR, with probability at least $1 - \delta$. How can this be achieved?
2. What is the expected computational cost of solving the problem described above, and how does it depend on the problem parameters?
3. How does the approach of question 1 compare with other existing methods?
4. How can this CVaR or VaR selection framework fit as part of an intelligence source decision model?

1.6 Contributions

Stochastic root finding is concerned with the problem of finding the roots of a function $f(x) = E_\theta F(\theta, x)$; that is, the expectation of a function F with a random vector θ. The primary techniques used to achieve this are sample average approximation and stochastic approximation [9]. Our work is the first that deals with the so-called *probably approximately correct* framework in a stochastic root finding setting, of which the value-at-risk and conditional value-at-risk are two critical cases. Our proof technique is based upon a coupling argument that seeks to obtain the bounds required to implement a probably approximately correct algorithm. There exist two papers that deal with quantiles in a probably approximately correct framework [10] and [11]; however, we have not discovered any papers that study stochastic root finding within the probably approximately correct framework.

1.7 Thesis Outline

This thesis is organized into five primary chapters. Following this introductory chapter, Chapter 2 discusses a background of the technical problem through the conduct of a literature review. Major topics in learning theory, as well a specific introduction to the multi-armed bandit, are given; a discussion of previous related work is also presented here. Chapter 3 depicts the mathematical algorithm derivations, as well linking the operational setting discussed in the preceding Chapter. Numerical analysis of the proposed algorithms is presented in Chapter 4, indicating the performance

of each algorithm in both high- and low-dimensional settings. The final chapter summarizes the totality of research that has been undertaken and looks toward the future in providing a defined way forward for further advancements in this research domain. For the non-technical reader, it is recommended that only the beginning of Chapter 3 be read and the remainder scanned.

CHAPTER 2:
Background and Literature Review

This chapter provides background on material relevant to the work presented in subsequent chapters. Beginning with a discussion on reinforcement learning, a broad link is made between a computational approach to learning and our problem specifically. We then investigate in depth the main problem for our research setting, the multi-armed bandit. We close with a discussion of contemporary literature that leads us to our contributions to the field, presented in Chapter 3.

2.1 Background

A number of approaches and techniques in use at present were born out of work from the last two centuries. Learning, from a computational perspective, is concerned with the actions taken by an agent in order to maximize a cumulative reward. Within computational machine learning there exist five key paradigms of learning: being supervised, unsupervised, online, active, and reinforcement.

1. **Supervised learning.** There are two main categories of algorithms within the supervised learning paradigm, consisting of classification and regression. The algorithms use a classic known dataset method with which to *train* the algorithm in making predictions. When used on test datasets that have no known properties, or for which we do not know anything about their properties, the algorithms use their knowledge from the training set with which to make predictions about the test set. Supervised learning is commonly used in such applications as financial credit risk analysis, algorithm based trading strategies and classifiers, and email spam filters. Nominally, we can use supervised learning in situations where we require *pattern recognition* of the data to be undertaken [12].
2. **Unsupervised learning.** The family of techniques under the banner of unsupervised learning use unlabelled data with which to make inferences, gain insights, and find patterns. Cluster analysis is the most systemic unsupervised learning method and is used to find hidden patterns in such data. Unsupervised learning algorithms are commonly used in applications such as data pattern mining, computer vision object recognition, and natural language processing [12].
3. **Online learning.** Within the online learning framework process, we attempt to answer a series of questions. In each iteration, we learn the answers to the previously posed questions without delay, this being the aforementioned online component, and is the distinguishing feature of this style of learning. Online systems see a systemic application in society. Such applications include recommender systems, where the *Netflix recommender problem* is a classic example

of this algorithm in use. Here, a user watches a film and provides immediate feedback to the system, feeding the algorithm and enabling further training of the recommender system as to what films the user may enjoy in the future. The notion of *regret* is introduced here as the difference between the system recommendations and the like or dislike of the user after viewing. We can measure the average success of the system in predicting a film for the user to watch and obtain a long-run appreciation of how it performs [13].

4. **Active learning.** As opposed to looking at the entire dataset, as was seen with supervised and unsupervised learning, the critical idea that distinguishes active learning is that it actively selects the training label subsets of the total dataset with which to select its data to learn from. Such problems arise from a framework where unlabelled data exists; however, there are further prohibitive reasons as to why the labels cannot be easily attained. Such reasons could include the cost of the labelled data, the time to manually to label the dataset or simply the labels are incomplete. The primary contrast to unsupervised learning is the ability to interactively query the user and obtain new data outputs. Active learning is commonly known as *query learning* or *optimal experimental design* in the machine learning literature, with applications in speech recognition, information extraction, and classification and filtering [14].
5. **Reinforcement learning.** Our final and most important computational learning concept (in the context of this thesis) is reinforcement learning. While it is commonly thought that reinforcement learning is a subset technique of unsupervised learning, this is not quite correct. Reinforcement learning is distinct as it tries to maximise a *reward*, such as in a Markov Decision Process, as opposed to the reliance on hidden structure, such as in unsupervised learning. The seminal problem in reinforcement learning is to maximise the *reward* of an *agent*, and as such leads us to the problem of *exploration vs exploitation* - a concept we will cover in detail. Within reinforcement learning the agent must *exploit* their current environmental knowledge to reveal a *reward*, whilst also *exploring* their surroundings in order to aid decision-making in the future. Pursuing a purely exploration or exploitation *policy* cannot be exclusively undertaken in the general setting and, as such, an efficient trade-off is required. Reinforcement learning considers the problem where an agent with specific goals interacts with an uncertain environment [15]. The paradigm of reinforcement learning is the setting we find ourselves in for the remainder of this thesis.

It is important to define a limited number of critical terms for use throughout the remainder of this thesis. The terms defined below have been adapted from [15].

1. **Learner (agent).** The learner, or agent as is also commonly known, is the subject who takes actions based on inputs from the environment in an attempt to maximize their observed reward.

2. **Policy.** A policy is defined as the way in which the learner behaves at a given iteration (time) step, based on the history of rewards earned and actions taken to-date.
3. **Reward.** In each iterative step, a (typically random) reward is received, which then influences the action taken in the next step. Earning rewards is the goal of the agent.
4. **Regret.** The regret is the difference between the reward that could have been earned with more complete information (see below) and the reward earned from the policy implemented. A good policy is one with regret grows slowly.

An important aspect of some learning problems is the trade-off seen between exploration versus exploitation. In a pure exploitation policy, the learner seeks to exploit the best of what is already known without considering alternative actions. When juxtaposed with a pure exploration policy, the learner attempts to take as many different actions as possible in order to make better selections in future iterations. While very specific exploitation-only and exploration-only algorithms exist, the overwhelming body of work that has been undertaken is in the development of hybrid algorithms to efficiently find this trade-off. The balance between an optimal action seen previously and exploring new actions at random iterations, according to a set policy, is the aim of these algorithms.

2.2 Multi-armed Bandits

The problem of multi-armed bandits first appeared in 1930s academic literature; however, it gained little traction in mathematical communities as it was thought that no closed form analytical solution to the problem existed. The introduction of the seminal paper by [16] set the framework for a reinvigoration of interest in the suite of now systemic multi-armed bandit problems. With the explosion of work to solve machine learning problems, the multi-armed bandit has seen consistent application towards this endeavor.

The multi-armed bandit is an iterative probabilistic decision learning problem in which, with a choice of k arms of a bandit available to the player at each discrete time step, a reward is observed by the player. The aim here is to select an arm to maximize the cumulative reward seen over a finite time horizon, or alternatively minimize the regret from the optimal selection possible.

For each time period t, an agent selects a single arm $k_t \in 1 \ldots K$ and receives the scalar reward $X_{k,t}$, where K is the number of arms. In the base case of the multi-armed bandit problem, we consider problems for which the reward $X_{k,t}$ is maximized. As derived in [17], the regret from the optimal selection after n rounds is defined as

$$R_n = \max_{i=1,\ldots,K} \sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}, \tag{2.1}$$

where I_t is the selected arm at time t, with an associated reward $X_{I_t,t}$. While a general notion of the concept of regret has been given, no formal definition has been provided. We define two key forms of regret, namely expected and pseudo-regret.

1. **Expected Regret.** The expected regret is the expected difference observed by an agent, with reference to an optimal action for the sequence of realized rewards [17].

$$E[R_n] = E\left[\max_{i=1,\ldots,K} \sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}\right]. \tag{2.2}$$

2. **Pseudo-Regret.** The pseudo-regret is a weaker form of regret, as an agent competes only against an optimal action, in expectation [17].

$$\overline{R_n} = \max_{i=1,\ldots,K} E\left[\sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t}\right]. \tag{2.3}$$

2.2.1 The Stochastic Multi-armed Bandit

The stochastic multi-armed bandit was initially presented by [16], introducing the technique to analyze upper confidence bounds for regret. The generalized form of the stochastic multi-armed bandit is defined in Algorithm 1; however, it should be noted that the underlying distribution of each arm does not change in each iteration. The reward observed in each time period is a random sample drawn from that arm's distribution [17].

Algorithm 1 The stochastic bandit problem

1: *Known parameters:* number of arms K and (possibly) number of rounds $n \geq K$.
2: *Unknown parameters:* K probability distributions $\nu_1, ..., \nu_k$ on [0,1]
3: For each round $t = 1, 2, ...$
   1. the forecaster chooses $I_t \in \{1, ..., K\}$;
   2. given I_t, the environment draws the reward $X_{I_t,t} \sim \nu_{I_t}$ independently from the past and reveals it to the forecaster.

The primary metric of interest for this family of algorithms is the pseudo-regret. The pseudo-regret of a stochastic multi-armed bandit is defined as a special form of the general pseudo-regret, given in Equation 2.3 as

$$\overline{R_n} = n\mu^* - \sum_{t=1}^{n} \mu_{I_t}, \tag{2.4}$$

where μ^* is defined as $\max_{i=1,\ldots,K} \mu_i$ and μ_{I_t} defined as the mean of arm I_t. An underpinning property of the stochastic multi-armed bandit is that it can be proven that a logarithmic upper bound determines the rate of convergence observed, given as $O(\log n)$, that cannot be improved upon. Equation 2.5 also can be expressed as

$$\overline{R}_n = \sum_{i=1}^{K} \Delta_i E T_i(n), \tag{2.5}$$

where $T_i(n)$ is the number of pulls of arm i by time n, and $\Delta_i = \mu^* - \mu_i, \forall i \in K$ [17].

2.2.2 Time Horizons

It is important to make a distinction between the infinite and finite time horizon cases. In the finite time case, we seek to select the optimal arm with a probability of at least $1 - \delta$, for a sufficiently small $\delta \in [0, 1]$. Alternative to this is the infinite time scenario, which is not considered in this thesis because time horizons are very much finite in the intelligence setting.

2.2.3 Key Algorithms

A number of foundational algorithms are critical to framing our work presented herein. These form the basis for the main body of research leading to our work presented in the subsequent chapters.

First, we consider the multi-armed bandit problem within the context of a probably approximately correct model. The first work is [18], who provide an algorithm to find the arm with largest expected reward with probability at least $1 - \delta$, where $\delta \in (0, 1)$ is a parameter selected by the agent. The successive elimination algorithm sequentially samples from the remaining candidate arms in each iteration, returning an observation and recalculating all summary values. At each time period, if an arm's empirical mean is sufficiently small, then it is removed from further consideration, thus reducing the feasible set of arms by one. For arms with distributions supported over $[-b, b]$, for $b > 0$, [18] shows that the expected number of observations until the algorithm terminates is of the order

$$64b^2 \log\left(\frac{K\pi^2}{6\delta}\right) \sum_{i \neq i^*} \Delta_i^{-2},$$

for a total of K arms, when the goal is to find an optimal arm with probability of at least $1 - \delta$. The authors show that such computational complexity is the lowest possible, up to the leading order. The lower bound on value based probably approximately correct bandit sample complexity was studied in detail within [19].

The algorithm derived by [20] appears below. The parameter α_n is the elimination threshold at

stage n, and depends on the arms distributions. For arms with support over $[-b, b]$, for $b > 0$, it is

$$\alpha_n = b\sqrt{\frac{2}{n}\log\left(\frac{Kn^2\pi^2}{6\delta}\right)}.$$

Algorithm 2 Successive elimination algorithm

1: Set $n = 1$ and $S = \{1, 2, ..., K\}$.
2: Set for each arm i, $\bar{X}_1(i) = 0$;
3: **R**epeat
- Sample every arm $i \in S$ once and let $\bar{X}_n(i)$ be the average reward of arm i by trials or pulls n;
- Let $\bar{X}_n(max) = \max_{i \in S} \bar{X}_n(i)$;
  - For each arm $i \in S$ such that $\bar{X}_n(max) - \bar{X}_n(i) \geq 2\alpha_n$ **do**
    * set $S = S - \{i\}$;
  - end
- $n = n + 1$;

4: **Until** $|S| > 1$;

Next, we discuss a sequential elimination algorithm closer to the focal problem of this thesis. The qualitative probably approximately correct `QPAC` algorithm (Algorithm 3) is an iterative adaptive elimination algorithm that probabilistically removes arms from consideration, based on the tests at lines 9 and 11 in the algorithm. This algorithm aims to select the arm with largest τ quantile, up to a resolution of ϵ (so-called (ϵ, τ)-optimal) with probability at least $1 - \delta$. The expected number of required samples required to determine the (ϵ, τ)-optimal arm with a probability of at least $1 - \delta$ is of order

$$O\left(\sum_{k=1}^{K} \frac{1}{(\epsilon \vee \Delta_k^{\epsilon})^2} \log \frac{K}{(\epsilon \vee \Delta_k^{\epsilon})^2 \cdot \delta}\right),$$

which is similar to that of Algorithm of 2. Thus, `QPAC` is shown to be optimal up to a logarithmic factor for the sample complexity.

Algorithm 3 relies on the empirical quantile

$$\hat{Q}_m^{X_k}(\tau) = \inf\{x \in R : \tau \leq \hat{F}_m^{X_k}(x)\},$$

where $\hat{F}_m^{X_k}(x)$, the empirical distribution of the rewards from arm k after m samples. The mathematical operator $\preceq$ indicates the totally ordered set with which the algorithm operates over, and c_t is an evaluated function value that depends on the iteration t (or alias sample m), given as an

auxillary function that determines the elimination confidence interval size, defined in Equation 2.6. The parameters x_t^+ and x_t^- are the thresholds for elimination that take the place of α_m in Algorithm 2, as follows:

1. if the value of the arm is less than x_t^-, it is removed from further consideration,
2. if the value of the arm is greater than x_t^+, it is selected as the optimal arm, exiting the algorithm,
3. if the value of the arm lies between x_t^- and x_t^+, it remains under consideration.

The parameters x_t^+ and x_t^- depend on constants defined as

$$c_m(\delta) = \sqrt{\frac{1}{2m} \log \frac{\pi^2 m^2}{3\delta}}. \tag{2.6}$$

This work leads us to Algorithm 3, in which we can note the underlying classic bandit framework described in Algorithm 1. For each iteration, a sample $X_{k,t}$ is drawn from every candidate arm in the set, followed by an update of the values x_t^+ and x_t^-, continuing until the candidate set is a singleton; we substitute m samples for t time-steps in the algorithm notation as we draw exactly one sample in each time step.

Algorithm 3 QPAC(δ, ϵ, τ)

1: Set $\mathcal{A} = 1, ..., K$ ▷ Active arms
2: $t = 1$
3: **while** $\mathcal{A} \neq \emptyset$ **do**
4: **for** $k \in \mathcal{A}$ **do**
5: Pull arm k and observe $X_{k,t}$
6: $x_t^- = \max_{k \in \mathcal{A}} \widehat{Q}_t^{X_k}(\tau - c_t(\frac{\delta}{K}))$
7: $x_t^+ = \max_{k \in \mathcal{A}} \widehat{Q}_t^{X_k}(\tau + c_t(\frac{\delta}{K}))$
8: **for** $k \in \mathcal{A}$ **do**
9: **if** $\widehat{Q}_t^{X_k}(\tau + c_t(\frac{\delta}{K})) \prec x_t^-$ **then**
10: $\mathcal{A} = \mathcal{A} \setminus \{k\}$ ▷ Discard k based on line 6
11: **if** $x_t^+ \preceq \widehat{Q}_t^{X_k}(\tau + c_t(\frac{\delta}{K}))$ **then**
12: $\widehat{k} = k$ ▷ Select k according to line 7
13: **BREAK**
14: $t = t + 1$
15: **return** $\widehat{k}$

2.3 Quantile and Superquantile Risk

The superquantile risk is a metric known within quantitative finance community as conditional value-at-risk, that quantifies the expected losses over a probabilistic threshold [3]. When a *pdf* exists, the superquantile is simply given as the conditional expectation above a given quantile threshold, stated as $E[X|X \geq q_\alpha]$, where X is the random variable corresponding to portfolio loss and q_α is the quantile threshold at the desired level of risk averseness α. That is, the conditional value-at-risk is the expected loss when the losses fall in the worst $1 - \alpha$ percentile. When $\alpha = 0$, there is an assumed agnosticism to risk, whereas when $\alpha = 1$, there is a complete averseness towards risk. Quantile risk, more commonly known as value-at-risk, is used as a systemic risk metric within the financial engineering community, and is given as the α quantile of the portfolio loss X [21]. The interpretation is that portfolio X has probability $1 - \alpha$ of incurring a loss of at least q_α. Figure 2.1, taken from [6], further illustrates the concepts described.

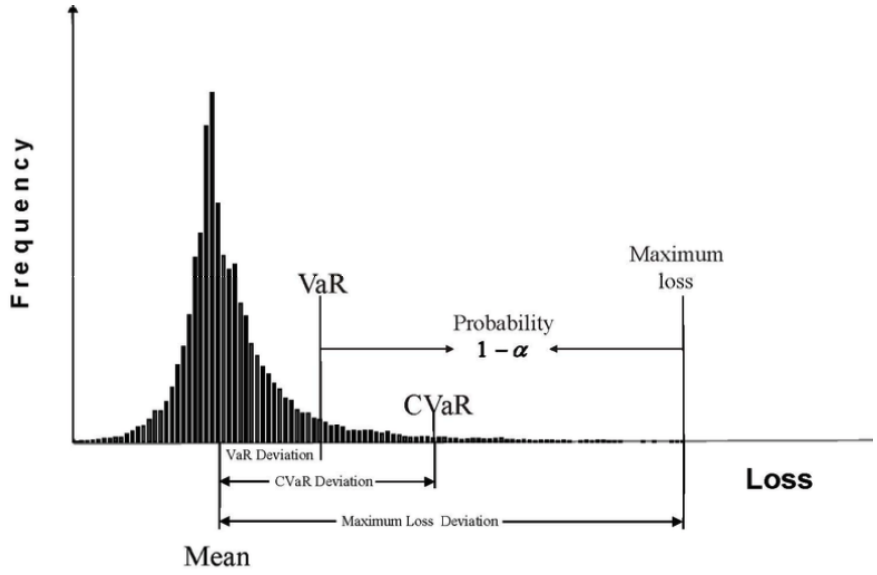


Figure 2.1. Illustration of Value-at-Risk and Conditional Value-at-Risk of the *pdf* of a Random Variable Y. Source: [6].

CHAPTER 3:
Computational Methods

3.1 Introduction

Following our background overview in the preceding chapter, we now turn to the model formulation and analysis.

Each arm represents an intelligence source, that produces an intelligence item per time period. The *importance* of an item is the value that is generated by sampling a source with respect to a specific request for information. The importance generated from a source at each time step is a distinct document observation, which we consider as a random variable, with the random variables being independent and identically distributed for each arm.

Recall from the last chapter our motivation: a request for information necessitates a certain average importance value—this being the conditional expectation superquantile—for the documents that are passed on to a senior analyst, meaning that a good source is one that has a large probability of producing such documents. More specifically, for each source, we set its conditional expected importance over a threshold (initially unknown) equal to the average document importance required (an input), and then seek to sample from the source whose quantile at the unknown threshold is smallest. Our measure of performance is the expected regret when compared to selecting the optimal source at each time step. We define the regret as the difference in quantiles between the best source and the suboptimal sources. In order to achieve this, our goal here is to produce algorithms with good regret convergence, which in our case means logarithmic in the number of documents explored.

Contained within Table 3.1 is a summary of the indexes, sets and variables mentioned within the description above.

Table 3.1. Parameters of the Model.

Constant	Index	Set	Variable	Description
	$s \in$	$\mathcal{S}$		Intelligence source s is a single element of the set of all sources $\mathcal{S}$
	$t \in$	$\mathcal{T}$		t is the current time period in the finite time horizon $\mathcal{T}$
			$X_{s,t}$	An observation from source s in period t
	$\alpha_s \in$	$(0, 1)$		The α-quantile of source s
	$k_{s,\alpha} \in$	(a, C)		The importance threshold of source s at quantile α. This value is found by the VaR algorithm
$C \in$		(a, b)		The desired intelligence request value

Overarching indexes, sets and variables used to describe our model of intelligence operations.

3.2 Selecting the Largest Quantile Risk Level

In this section, we present the general model. We study the problem faced by an analyst who has a constraint on the superquantile risk for a set of candidate intelligence sources, meaning that

$$E[X_s | X_s > k_{s,\alpha_s}] = C, \tag{3.1}$$

where X_s is a random regret associated with an intelligence source s, and k_{s,α_s} is the quantile risk at level α_s; $C \in \mathbb{R}$ is the input constraint.

The constant C (a model input) is the average importance of the intelligence for the documents that are passed to a senior analyst. $1 - \alpha_s(C)$ is the fraction of documents generated by source s that have a conditional expected importance of C, with quality at least $k_s(C)$ (unknown). For a given C, the goal is to find the intelligence source that produces the largest fraction of items that meet the intelligence quality level and hence, the analyst wishes to find the source with lowest $\alpha_s(C)$. In this case, a sample $X_{s,t}$ is the quality of source S generated in time step t. Later in this section we impose the condition that $E[X_s] < C$, corresponding to the average quality of a source generated by source s being less than C; otherwise $\alpha_s(C)$ is 0, meaning that all items are passed for further analysis (and there is no problem to solve). As C is a qualitative constant, the calculation of means, medians and variance on an ordinal set of observations is an invalid measure and cannot produce meaningful outcomes [10]. Critically, we want to solve for $\alpha_s(C)$, which requires first solving for the value of $k_{s,\alpha}$, which is the root problem. There are two important cases to consider:

1. **A high C.** Relative to the interval (a, b), a high C indicates an important intelligence request. Here, we wish to find the source(s) that generate items of high average importance. In this scenario, it is likely that each source generates relatively few items over the threshold $k(C)$ so that finding the source for which $P(\text{item importance} > k(C))$ is largest is realistic. The latter is akin to finding the source with the highest $\alpha_s(C)$; see the definition 3.14.
2. **A low C.** Relative to the interval (a, b), a low C indicates that this is a less important intelligence request. This is the converse of the preceding scenario. Here the flow of items is likely to be large, and the analyst is interested in finding the source for which $P(\text{item importance} > k(C))$ is smallest. Appealing to (3.14), this means that $\alpha_s(C)$ is smallest.

The solution of the root problem also is called the *buffered Probability of Exceedance* (bPOE) [22], defined as the inverse of a conditional value-at-risk level and is a generalization of the *buffered Probability of Failure* (bPOF), defined in [23] as one minus the inverse at point zero of the superquantile or conditional value-at-risk level.

We consider a framework with two cases. First, the analyst wishes to find the intelligence source with largest or smallest threshold quantile (known as the value-at-risk or quantile at level α), given as the root of the equation k_{s,α_s} in Equation 3.1. Second, the analyst's objective is to identify the intelligence source with largest or smallest probability α of exceeding the quantile risk, where the root k_{s,α_s} is set to satisfy Equation 3.1. In the former case, the level α_s plays no role in finding the quantile risk that satisfies the superquantile risk constraint, while in the latter α_s can be obtained from the root k_{s,α_s}.

Without loss of generality, we work with the problem of finding the source(s) with the largest root k, as well as the one with largest superquantile risk level α. The problem of finding the arm with the smallest superquantile risk level or root is solved by utilising our multi-armed bandit model arms driven by the negative random variables $-X_s$. Note here that the problem of finding the root of $C' = E[X'_s | X'_s \leq k]$ with $E[X'_s] > C'$ and $P(X'_s < C') > 0$ is identical to the situation considered here by allowing $X = -X'$ and $C = -C'$.

More formally, we consider a finite set of candidate arms $\mathcal{S} = \{1, \ldots, S\}$. For each arm $s \in \mathcal{S}$ there is a stochastic observation, defined by a random variable X_s. For the purposes of our model, we assume that X_s has a continuous distribution, and thus a density, for each arm $s \in \mathcal{S}$. The analyst observes independent and identically distributed (*iid*) samples $X_{s,1}, X_{s,2}, \ldots, X_{s,n}$ from a

distribution with a density $f_{X_s}(\cdot)$, and where $k_s(C)$ is the root of

$$C = E[X_s | X_s > k]. \tag{3.2}$$

The goal is to find the arm $s^* \in \mathcal{S}$ with the largest root, $k_{s^*}(C) = \max_s k_s(C)$.

Three key assumptions are made going forward for the remainder of this work, which are:

A1. $C - E[X_s] \geq \gamma > 0, \forall s \in \mathcal{S}$.

A2. The random variables $X_{s,1}, X_{s,2}, \ldots, X_{s,n}$ have bounded support over (a, b), $\forall s \in \mathcal{S}$, with $-\infty < a < C < b < \infty$.

A3. The random variables X_s, for all $s \in \mathcal{S}$, have a probability density function $f_{X_s}(\cdot)$ that is uniformly bounded below by ζ, governed by the constraint $\zeta > 0$.

Assumptions *A1* and *A2* ensure that the root $k_s(C)$ is well defined, whilst Assumption *A3* is used to bound the error probability of the root estimator. In quantile estimation settings, a positive density is required in the neighborhood of the quantile to control the estimation error; in the superquantile risk setting, this assumption is further extended to the entire support [24].

For the remainder of this work, we drop the arm index (source) s, unless required to depict a specific scenario for distinct arms. We turn to the work of [18], where our idea is to adapt a sequential elimination approach for which one needs to show that for $0 < \delta < 1$ there exists $\epsilon_n > 0$ such that

$$P(|k_n - k(C)| > \epsilon_n) \leq \frac{6\delta}{\pi^2 n^2 S}, \tag{3.3}$$

where k_n is the root estimator using n *iid* samples. The analysis is further simplified by exclusively dealing with the root of the function

$$g(k) = E[(X - C)I(X > k)], \tag{3.4}$$

which is the result of a simplification of

$$C = E[X | X > k] = \frac{E[X; X > k]}{P(X > k)} \iff g(k) = 0. \tag{3.5}$$

Assumptions *A1* and *A2* provide guarantees that $\lim_{k \to -\infty} g(k) = E[X] - C < 0$, and $g(\cdot)$ increases to attain its maximum at $k = C$, with $g(C) = E[(X - C)I(X > C)] > 0$. After attaining this maximum point, $g(k)$ monotonically decreases towards 0, as k approaches b. It follows that there

is only one root $k(C) < C$ that solves $g(k) = 0$. Of note, a consequence of Assumption *A3* is that

$$C - k(C) \geq \psi > 0 \tag{3.6}$$

for some $\psi > 0$; see Lemma 1 in Appendix A.1 for our proof of this. Intuitively, the error probability for the root estimation grows as C approaches $k(C)$ for a given sample size n; an illustration of the function $g(\cdot)$ is shown in Figure 3.1 that depicts this described function.

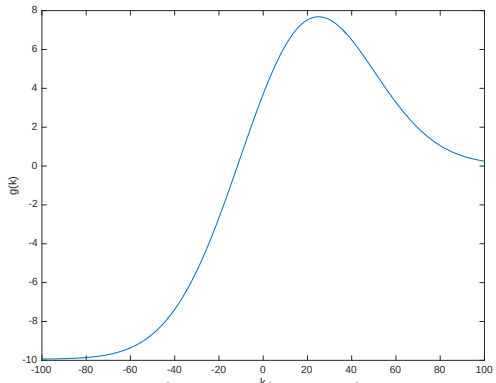


Figure 3.1. $g(\cdot)$ for a Truncated Normal Distribution ($\mu = 15, \sigma = 30$), Over the Interval $(-100, 100)$ with $C = 25$.

At each iteration of our sequential elimination algorithms, we draw *iid* samples $X_1, \ldots, X_n$, where the root is estimated by solving

$$\frac{1}{n} \sum_{i=1}^{n} (-C)\, I\,(X_i > k) = 0, \tag{3.7}$$

and the left-hand side of Equation 3.7 is interpreted as an empirical $g(\cdot)$ function. Moreover, we let the estimated root to be given by

$$k_n = \inf \left\{ k \geq a : \frac{1}{n} \sum_{i=1}^{n} (X_i - C)\, I\,(X_i > k) \geq 0 \right\}. \tag{3.8}$$

There exist three cases, which are

1. $(1/n) \sum_{i=1}^{n} X_i < C$ and $(1/n) \sum_{i=1}^{n} I\,(X_i > C) > 0$, in which case monotonicity of $(1/n) \sum_{i=1}^{n} (X_i - C)\, I\,(X_i > k)$ in k ensures that there is a unique root in (a, C).
2. $(1/n) \sum_{i=1}^{n} X_i \geq C$, leading to $k_n = a$.
3. $(1/n) \sum_{i=1}^{n} I\,(X_i > C) = 0$, in which case $k_n = \max_{i=1,\ldots,n} X_i \leq C$.

These assumptions imply that the probability of Cases 2 or 3 decay to zero exponentially in n.

In Case 1, given the ordered samples $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$, root finding can be equivalently implemented as $k_n = X_{(m^*)}$, where

$$m^* = \min\left\{ m \geq 1 : \sum_{i=m}^{n} \left(X_{(i)} - C\right) \geq 0 \right\}. \tag{3.9}$$

The average complexity of sorting the samples and finding $X_{(m^*)}$ is of order $O\left(n \log n\right)$, as given in [25].

3.2.1 Algorithm

We let $\Delta_s = k_{s^*}(C) - k_s(C) > 0$ for all arms $s \neq s^*$. Algorithm 4, the sequential quantile elimination algorithm shown below, initializes each root $k_{s,n}$ to a, and utilizes the threshold

$$\epsilon_n = \left(\log\left(\frac{\pi^2 n^2 S}{3\delta} \right) \frac{1}{2n} \right)^{1/2} \frac{b-a}{\zeta\psi}, \text{ for } n = 1, 2, \ldots, N, \tag{3.10}$$

to eliminate non-optimal multi-armed bandit arms (recall Equation 3.6 for the definition of ψ.) Theorem 1 shows that the root estimation error $|k_{s,n} - k_s(C)|$ is larger than ϵ_n, with a probability of δ. Algorithm 4 is a standard implementation of the sequential elimination algorithm of [18], with modifications as shown.

Theorem 1 *Under Assumptions* A1*,* A2*, and* A3*,*

$$P\left(|k_{s,n} - k_s(C)| \leq \epsilon_n, \forall n, \forall s = 1, \ldots, S\right) \geq 1 - \delta. \tag{3.11}$$

Algorithm 4 Sequential Quantile Elimination Algorithm $(C, \sigma, a, b, S, \delta)$

```
Set A = {1, ..., S}.
Set k_{s,n} = a, ∀s ∈ A
while |A| > 1 do
    for arm s ∈ A do
        Draw one sample from arm s and compute k_{s,n}
        if k_{max,n} = max_{s'∈A}{k_{s',n}} − k_{s,n} > 2ε_n then
            A = A \ {s}
    Set n = n + 1
```

Since $P(|k_{s,n} - k_s(C)| \leq \epsilon_n) \geq 1 - \delta$, Algorithm 4 probabilistically selects the best multi-armed bandit arm with a probability of at least $1 - \delta$, shown in [18]. Additionally, the work presented

in [20] proves that the expected number of samples $E[N_s]$ generated by a non-optimal arm $s \neq s^*$ is given as

$$\begin{aligned} E[N_s] &\leq \sum_{n=1}^{\infty} P(k_{\max,n} - k_{s,n} < 2\epsilon_n) \\ &\leq \sum_{n=1}^{\infty} P(k_{s^*,n} - k_{s,n} < 2\epsilon_n) \\ &\leq u_s + \sum_{n=u_s+1}^{\infty} P(k_{s^*,n} - k_{s,n} < 2\epsilon_n), \end{aligned} \tag{3.12}$$

for $u_s = \inf\{n : 4\epsilon_n \leq \Delta_s\}$. It easily follows that $E[N_s] \leq u_s + 2\delta/S$, concluding that the expected number of required samples for the non-optimal arms, $\sum_{s\neq s^*} E[N_s]$, is at most $2\delta + \sum_{s\neq s^*} u_s$. By solving for for $n \geq e$ such that $4\epsilon_n = \Delta_s$, with ϵ_n as in Equation 3.10, leads us to the dominant term in $\sum_{s\neq s^*} E[N_s]$, which is

$$\frac{8(b-a)^2}{\zeta^2\psi^2} \log\left(\frac{\pi^2 S}{3\delta}\right) \sum_{s\neq s^*} \Delta_s^{-2}, \tag{3.13}$$

for any choice of δ, given δ is small; see [20] for rigorous proofs of this. As a result of these constraints, when relative to the more traditional problem of finding a bandit arm with the largest expected value, finding a bandit arm with the largest root increases the expected number of required observations by a factor of $1/(\zeta\psi)^2$.

3.3 Selecting the Largest Superquantile Risk Level

We now return to our initial problem scenario of finding the source with the superquantile risk level. Moreover, for $F_s(\cdot)$ the distribution function of X_s, we let

$$\alpha_s(C) = F_s(k_s(C)), \tag{3.14}$$

where $k_s(C)$ is the root of

$$E[X_s | X_s > k] = C. \tag{3.15}$$

The analyst's goal here is to find the arm s^* with largest $\alpha_s(C)$. The empirical estimator of $\alpha_s(C)$ is $\alpha_{s,n}$, defined as

$$\alpha_{s,n} = \frac{1}{n} \sum_{i=1}^{n} I(X_i \leq k_n), \tag{3.16}$$

where k_n is as defined in Equation 3.8 for arm s. It follows from Equation 3.9 that $\alpha_{s,n} = m^*/n$ when $(1/n) \sum_{i=1}^{n} X_i < C$ and $(1/n) \sum_{i=1}^{n} I(X_i > C) > 0$, where m^* is as in Equation 3.9. Hence, this problem is computationally not any costlier than that of finding the root k_n.

Toward the goal of deriving a sequential elimination algorithm, we define the threshold

$$\epsilon_n = \left(\log\left(\frac{\pi^2 n^2 S}{\delta}\right)\frac{2}{n}\right)^{1/2} \times \max\left\{\frac{b-a}{\psi^2\zeta}, \frac{2(b-a)+(b-C)/n)}{\psi}, 1\right\}, \tag{3.17}$$

for $n = 1, 2, \ldots, N$. The maximand for Equation 3.17 arises as a result of coupling the empirical distribution to the empirical $g(\cdot)$ function (cf., Equation 3.7); see the proof of Theorem 2 given at Appendix A.2.

Intuitively, the difference between the true and empirical superquantile risk levels is large if at least one of three of the following events occur

1. the true root estimator significantly deviates from the true root,
2. the empirical $g(\cdot)$ function at the true root $k(C)$ significantly deviates from $g(k(C))$, or
3. the empirical distribution significantly deviates from the true distribution, at the root $k(C)$.

As with Section 3.2, we assume $\Delta_s = \alpha_{s^*}(C) - \alpha_s(C) > 0$ for all arms $s \neq s^*$. Algorithm 5 utilizes the thresholds in Equation 3.17 to eliminate non-optimal arms. Theorem 2, whose proof appears in Appendix A.2, proves the key step for our algorithm to function as prescribed.

Theorem 2 *Under Assumptions* A1*,* A2*, and* A3*, for ϵ_n as in (3.17) we obtain that*

$$P\left(|\alpha_{s,n} - \alpha_s(C)| \leq \epsilon_n, \forall n, \forall s = 1, \ldots, S\right) \geq 1 - \delta. \tag{3.18}$$

3.3.1 Algorithm

As a result of Theorem 2, we now present the sequential elimination algorithm for a superquantile risk level selection.

Algorithm 5 Sequential Superquantile Risk Level Elimination Algorithm $(C, \sigma, a, b, S, \delta)$

Set $\mathcal{A} = \{1, \ldots, S\}$.
Set $\alpha_{s,n} = 0, \forall s \in \mathcal{A}$
while $|\mathcal{A}| > 1$ **do**
  for arm $s \in \mathcal{A}$ **do**
    Sample from arm s and compute $\alpha_{s,n}$
    if $\max_{s' \in \mathcal{A}}\{\alpha_{s',n}\} - \alpha_{s,n} > 2\epsilon_n$ **then**
      $\mathcal{A} = \mathcal{A} \setminus \{s\}$
  Set $n = n + 1$

As with Algorithm 4, Theorem 2 implies that the arm with largest superquantile risk level α is chosen with a probability of at least $1 - \delta$. Furthermore, the expected number of total samples $\sum_{s \neq s^*} E[N_s]$ observed by the non-optimal bandit arms is given to us by

$$\sum_{s \neq s^*} E[N_s] \leq 2\delta + \sum_{s \neq s^*} u_s \tag{3.19}$$

for $u_s = \inf\{n > e : 4\epsilon_n \leq \Delta_s\}$. For small $\delta > 0$ and by standard arguments, we see that the dominant term in $\sum_{s \neq s^*} E[N_s]$ is

$$32 \left(\max \left\{ \frac{b-a}{\psi^2 \zeta}, \frac{2(b-a)}{\psi}, 1 \right\} \right)^2 \log \left(\frac{\pi^2 S}{\delta} \right) \sum_{s \neq s^*} \Delta_s^{-2}, \tag{3.20}$$

where the impact of the $(b - C)/n$ term showing in Equation 3.17 is of an order that is smaller than that of $\log(1/\delta)$.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Numerical Examples

Following the derivations provided in the preceding chapter, presented here are numerical examples for three primary distributions: the truncated normal, triangular, and uniform. We begin with a validation of our derivations and show that we indeed do elicit numerically accurate estimates for our root function $g(\cdot)$. Following this, we juxtapose each algorithm against each distribution, with a brief introduction to the parameters selected for the remainder of this chapter. At the conclusion of these numerical examples, implementations are presented for extended length trials, specifically investigation of long-run and high sample trials. In combining these concepts together, a high dimensional data section has been included that looks at the scalability of these algorithms up to a $10^8 \times 10^2$ matrix. We finish by providing an analysis of ϵ and investigate the convergence of this threshold, as well its effect on the rate of elimination for each algorithm.

4.1 Implementation

To contrast the effect that different distributions have on the rate of convergence, the input parameters for each algorithm remained constant throughout each implementation of both the quantile and superquantile elimination algorithms within this chapter and defined in Chapter 3. These parameters we selected so as to illustrate the effect of convergence in a sufficient number of iterations. The inputs to both the quantile and superquantile elimination algorithm are identical, with the mean of each arm, μ, calculated based upon the interval of the underlying distribution, $[a, b]$. Here, all arm means are set to be S equally linearly spaced values, over the interval $[a, b]$, for all distributions. The number of new observations considered in each iteration, n, has been set sufficiently large in order to ensure that a timely convergence occurs. This represents the consideration of multiple source items at each iteration, as opposed to selecting only a single item. While this assumption may not hold in all instances for an on-line implementation, it provides us with the numerical convergence properties we seek to demonstrate here. Note that in Table 4.1 that no information on the distribution of each arm is given. For the numerical examples presented in later sections, we consider each arm as having the same underlying distribution, however, with distinct μ in order to observe convergence. Note that the standard deviation, σ remains constant for each arm of the bandit where μ varies. We have not mixed distributions for arms.

We note that the depiction of both the quantile and superquantile elimination algorithms in Chapter 3 force the algorithm to continue until convergence has been achieved. However, for the purposes of practical implementation, we provide an upper bound on the number of allowed iterations, *max*

iterations. This parameter is a practical bound that affords us the opportunity to exit the algorithm and observe the rate of elimination at distinct iterations of the algorithm. Note that if the algorithm fully converges prior to the attainment of this bound, the standard stopping criteria will execute as given in Chapter 3.

At the beginning of the algorithm, as shown in Appendix A.3 and A.4, the parameters C, σ, a, b, S, and δ are used to calculate the mean of each arm (μ), as well to construct the data structure with to operate on. We pull n observations from each arm, drawn as random samples of each arm's underlying probability distribution, calculate our elimination criteria, and determine for each arm if it is eliminated or selected as optimal. If neither of these cases is met, the arm remains in consideration until the next iteration. This road map for the execution of the algorithm is identical for all arms within the system.

Seen within Appendix A.3 and A.4 are the outputs listings from these algorithms, updated at every iteration. The `result` matrix is a series of S vectors that contain the empirical estimate of our root function, $g(\cdot)$; this is value of each arm that is shown with Figures 4.1 to 4.4. `epsilon` is a vector of the values of ϵ and was the data used to depict Figures 4.11 and 4.12. `verbose_arms` is a vector that tracks the status of each arm and indicates which arms are currently active or that have been eliminated. The calculated means μ—discussed above—are contained within the vector `mu` and is a vector of S linearly spaced values used for each arm, throughout. The vector `root_max` tracks the best-observed arm in each iteration. Within the limit, this vector will depict the optimal arm continuously, whereas a number of arms are selected initially as the primary metric, `result`, stabilizes. The final parameter recorded is a vector of the remaining number of expected values required for convergence, `expected_samples`.

The parameters are selected in order to provide a visually striking difference for each distribution presented in this chapter are

Table 4.1. Parameters Used for Numerical Examples.

Parameter	Value
C	25
σ	30
a	-100
b	100
S	25
δ	0.1
n	10^4
max iterations	500

Where these parameters have not been used, this is specifically stated. These parameters represent our base case scenario for evaluation; max iterations is the algorithm stopping criterion.

4.1.1 Code Development

Initially, each component of the algorithm was individually implemented and tested, where knowing the theoretical results for the truncated normal case helped us to verify the code. Program builds first occurred within the *R-3.3.2* environment, providing a good foundation for rapid development. Subsequently, a migration to the *MATLAB-R2016b* and then *MATLAB-R2017a* platforms was made, occurring for two primary reasons

1. the ability to incorporate specific library files, and
2. for implementation on *Hamming*. The Naval Postgraduate School has a high-performance computing system, in the form of a hybrid cluster supercomputer; this is the Hamming system. All numerical execution was conducted on this architecture.

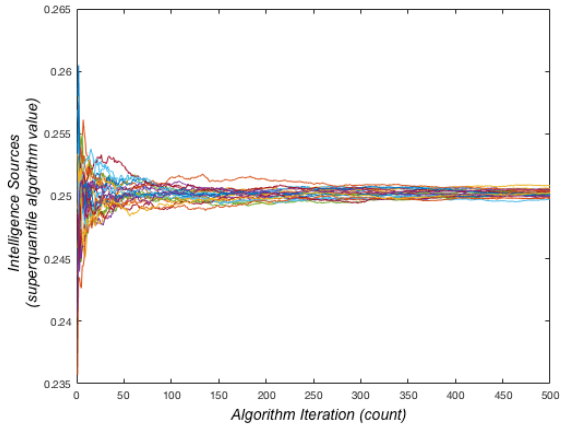
Upon obtaining numerically stable implementations, the modular system was discarded for a streamlined single-function script, reducing node communication requirements between modules. The final version of the implemented code for both the VaR and CVaR sequential elimination algorithms can be seen in Appendix A.3 and A.4.

4.1.2 Truncated Normal Distribution

During scoping of this topic for research, we had intended to implement both algorithms in the case of unbounded support through the use of distributions such as the classic Normal. This concept

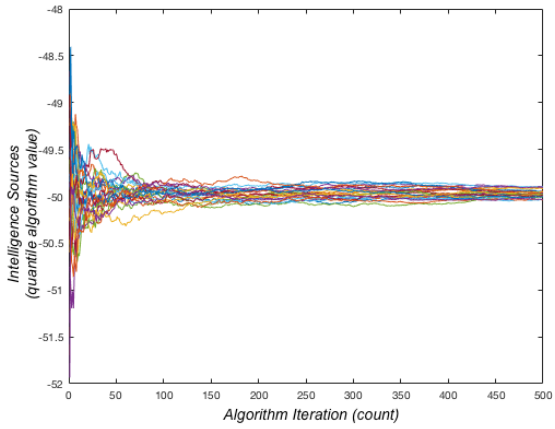
was refined early on to only consider the case of bounded distributions and as a result, we use the truncated normal as our base case scenario. Importantly, this still allows for the derivation of closed-form solutions to be undertaken and ensure that we do not become overly dependent on only the numerical implementation of our measurement and assessment.

The truncated normal offers the lowest rate of arm elimination for our algorithm. When we compare Figures 4.1 and 4.2 with those in each column (Figures 4.3 to 4.6), it is evident that there is a greater number of remaining arms at the termination of the algorithm than for both the triangular and uniform cases; this will be discussed in detail in later sections. At this resolution, not one bandit arm has been eliminated thus far. In a dense setting such as the one provided, it is not possible to see this non-elimination, thus far. In order to note the proportion of eliminated arms, we interrogate one of our output vectors which track the elimination of arms. While we observe convergence to the mean of each arm—when compared to theoretical results derived in Chapter 3—a far greater number of iterations is required to reach full algorithmic convergence and identify the correct arm with a probability of $1 - \delta$. This happens because the ζ value is very small (see below), meaning that the thresholds ϵ_n (of order $1/\zeta$) are large. The numerical examples suggest that the thresholds ϵ_n as presented in the previous chapter are too conservative. Figure 4.1 is a depiction of the standard setting for 500 iterations, with an underlying truncated normal distribution. The value of the y-axis is the quantile associated with the threshold C. When juxtaposed to Figure 4.2, we note that while the elimination example and convergence properties are similar, the y-axis has a strikingly different scale. In this instance, we are dealing with the basic quantile setting and as such, the y-axis represents the raw value, $X_{s,t}$. Each line in Figures 4.1 to 4.6 indicates a distinct arm (or source) which is under consideration, as given in the model description in Chapter 2. The value of each arm in every iteration is the solution to our empirical root equation $g(\cdot)$, described in the previous chapter.



CVaR Elimination for the Truncated Normal.

Figure 4.1. Implementation of Algorithm 5.



VaR Elimination for the Truncated Normal.

Figure 4.2. Implementation of Algorithm 4.

Regarding the derivation of the $g(\cdot)$ function for the truncated normal, we proceed as follows. We solve for k in

$$0 = E[(X - C)I(X > k)] = E[XI(X > k)] - CP(X > k). \tag{4.1}$$

The CDF of X is distributed as a truncated normal between a and b, with mean $\mu \in (a, b)$ and variance σ^2 is

$$P(X \leq k) = \frac{\Phi(\frac{k-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}$$

for k between a and b, ϕ and Φ, are the pdf and CDF of a standard normal distribution ($\mathcal{N}(0, 1)$), respectively. Hence,

$$P(X > k) = 1 - \frac{\Phi(\frac{k-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} = \frac{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{k-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}$$

The pdf is given as

$$\frac{1}{\sigma} \frac{\phi(\frac{k-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})},$$

and the ζ values used in the algorithm are capped at

$$\zeta = \frac{1}{\sigma} \frac{\min\{\phi(\frac{a-\mu}{\sigma}), \phi(\frac{b-\mu}{\sigma})\}}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}.$$

For the second term in Equation 4.1

$$\begin{aligned} E[XI(X > k)] &= \int_k^b x f(x) dx \\ &= \frac{\frac{1}{\sigma} \int_k^b x\phi(\frac{x-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} \\ &= \sigma \left(\phi\left(\frac{k-\mu}{\sigma}\right) - \phi\left(\frac{b-\mu}{\sigma}\right) \right) + \mu \left(\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{k-\mu}{\sigma}\right) \right). \end{aligned}$$

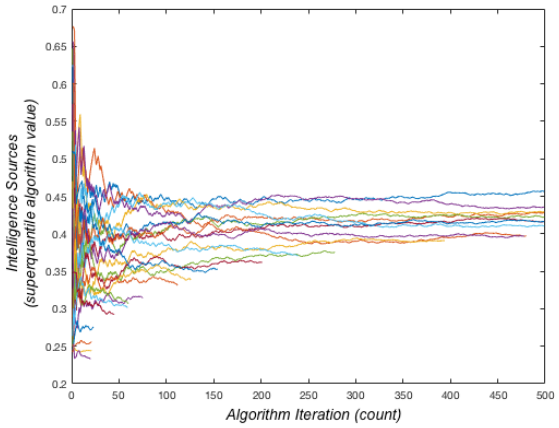
From here, $0 = E[XI(X > k)] - CP(X > k)$, which leads to

$$0 = \sigma \left(\phi\left(\frac{k-\mu}{\sigma}\right) - \phi\left(\frac{b-\mu}{\sigma}\right) \right) + \mu \left(\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{k-\mu}{\sigma}\right) \right) - C(\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{k-\mu}{\sigma})), \tag{4.2}$$

which is solved numerically (in MATLAB) for the value of k, which we have defined as the function $g(\cdot)$ within the Chapter 3.

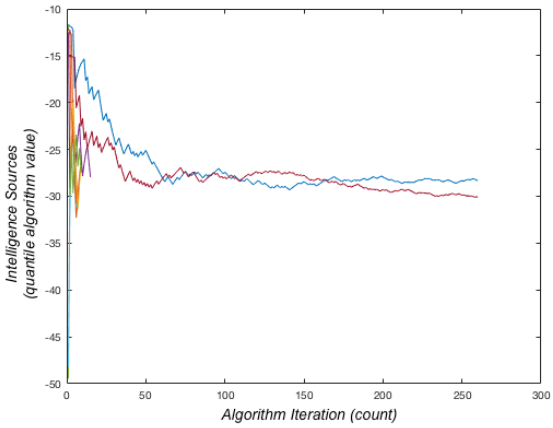
4.1.3 Triangular Distribution

Our second example is that of a modified triangular distribution, with parameters used as given in Table 4.1. To ensure numerical stability at the end points of the algorithm when eliminating arms from contention as optimal, the parameters ψ and ζ require strictly positive values. This requires a modified triangular distribution that sits on top of a uniform distribution to ensure we do not enter the case of $\psi \leq 0$ or $\zeta \leq 0$. The triangular distribution represents the intermediate case of convergence for our algorithms. This is to be expected, as ζ is constant for the uniform case and too small in the truncated normal scenario. We note that both algorithms very quickly eliminate arms from consideration. While the Algorithm 5 has not yet converged at 500 iterations, convergence was observed for Algorithm 4 in just over half of the maximum number of allowed iterations. Figure 4.3 is a depiction of the standard setting (given in Table 4.1) for 500 iterations, with an underlying triangular distribution. As with the previous section, the value of the y-axis is the quantile associated with the threshold C. When juxtaposed to Figure 4.4, we note that while the elimination example and convergence properties are similar, however, the y-axis has a strikingly different scale. In this instance, we are dealing with the basic quantile setting and as such, the y-axis represents the raw value, $X_{s,t}$



CVaR Elimination for the Triangular Distribution.

Figure 4.3. Implementation of Algorithm 5.



VaR Elimination for the Triangular Distribution.

Figure 4.4. Implementation of Algorithm 4.

4.1.4 Uniform Distribution

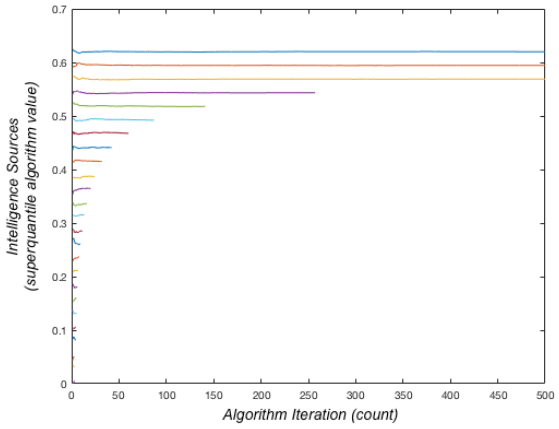
The final distribution under consideration is the uniform. This example represents the best-case convergence of all distributions used to illustrate each algorithm. The bounds for each uniform are given by

$$\left(\mu_s - \frac{b-a}{2}, \quad \mu_s + \frac{b-a}{2}\right), \qquad \forall \quad b > a,$$

where μ_s is obtained from the input parameter S, with all other parameters are as per Table 4.1; see Algorithm 4 at Appendix A.3, and Algorithm 5 at Appendix A.4. This provides the necessary difference for each algorithm to differentiate between S uniforms, with constructed linearly spaced means.

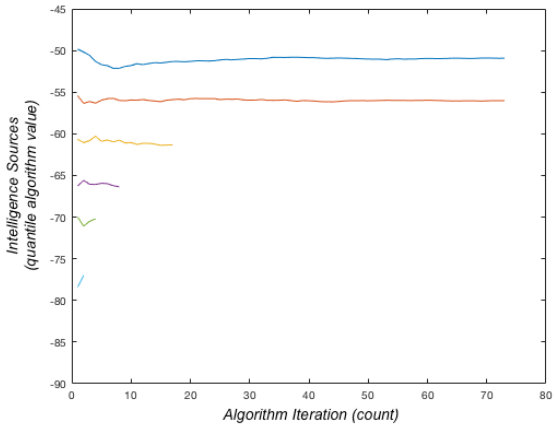
We observe here the same elimination trend as was discussed previously for the triangular distribution: the VaR elimination algorithm liberally eliminating arms, whereas the CVaR elimination algorithm retains arms for further consideration, for longer periods. In striking contrast, it would appear that two separate number of arms are under consideration in Figures 4.5 and 4.6, however, this not the case. As the convergence for the uniform is the fastest of our three examples, we observe a rapid elimination of arms in both algorithms, and near instantly in the case of the VaR elimination algorithm. Relatively few observations are required for this algorithm to discard arms that are non-optimal. What we observe in the limit of this execution is the arm with the largest $g(\cdot)$ value (i.e., the highest line).

As with both of our previous cases, the CVaR elimination algorithm was unable to successfully converge in the given number of iterations, with 3⁄25 arms remaining in contention. As will be seen in the following section, the elimination of the majority of non-optimal arms occurs quite quickly; however, convergence to the optimal with the last few remaining arms is where the majority amount of time is spent for each algorithm. Figure 4.5 is a depiction of the standard setting for 500 iterations, with an underlying uniform distribution. The value of the y-axis is the quantile associated with the threshold C. When juxtaposed to Figure 4.6, we note that while the elimination example and convergence properties are similar, however, the y-axis has a strikingly different scale. In this instance, we are dealing with the basic quantile setting and as such, the y-axis represents the raw value, $X_{s,t}$



CVaR Elimination for the Uniform Distribution.

Figure 4.5. Implementation of Algorithm 5.



CVaR Elimination for the Uniform Distribution.

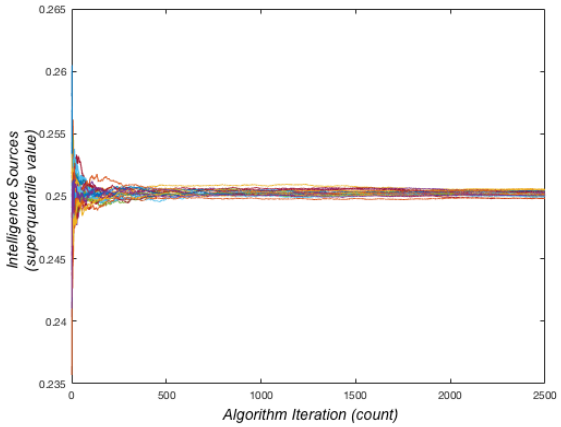
Figure 4.6. Implementation of Algorithm 4.

4.2 Extended Length Implementation

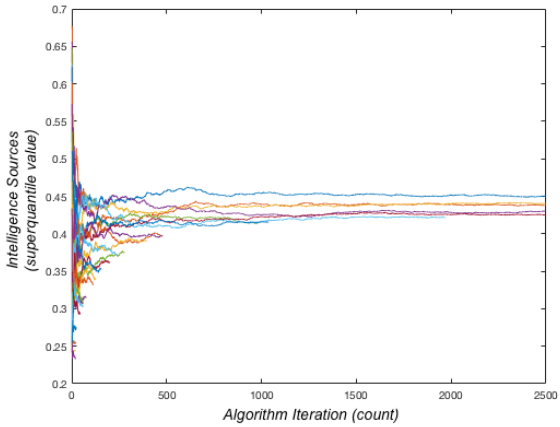
An investigation of the effect of increasing the number of allowed iterations, as well as the number of samples in each iteration, was undertaken. The aim of this analysis was to seek to understand the algorithmic behavior in the numerical limit of the algorithms.

4.2.1 Long-Run Trials

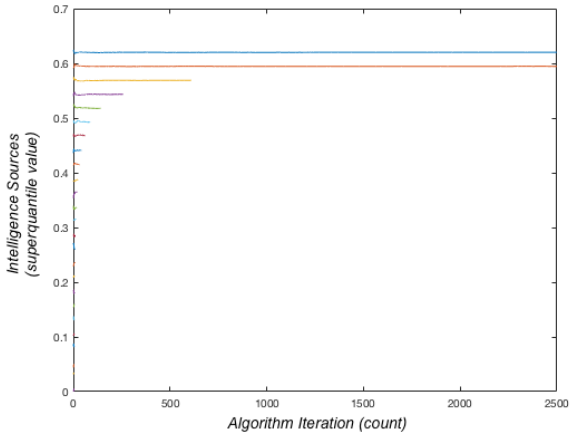
To explore algorithm performance with a greater number of iterations, each of the CVaR elimination algorithms was executed with only a single modification from the standard parameters: the maximum number of allowed iterations, being $2,500$. While convergence to a single optimal arm was again not observed, we can note that fewer arms are remaining for consideration at this point. This clearly indicates the more extreme case of the logarithmic expected number of iterations derived in the previous chapter. While no official timing was undertaken, the time for each algorithm to execute 500 iterations was approximately 0.75 hours, whereas the time taken to execute $2,500$ iterations was approximately 41 hours: a super-linear increase in the time required for each subsequent iteration to complete. At the completion of this algorithm for each distribution, 25,000,000 samples had been used to create the data in figures 4.7(a) to 4.7(c). From our derivation in the previous chapter, it has been calculated that approximately 5,800,000 more samples for each distribution would be required to reach full convergence to the optimal arm(s), with a probability of at least $1 - \delta$, where $\delta = 0.1$. This estimation is given from the derivation of Theorem 2.



2500 iterations of $n = 10^4$ samples for each iteration.



Truncated Normal (a) - left and Triangular (b) - right.



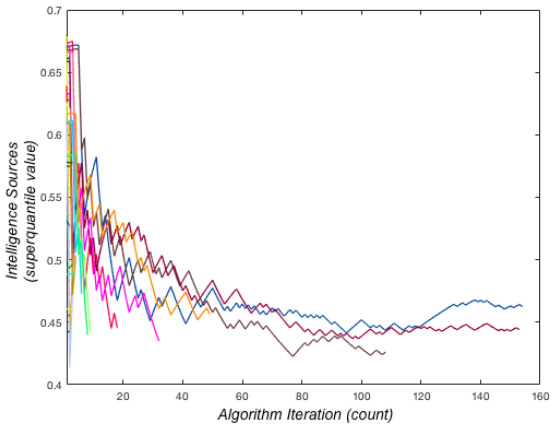
Uniform distribution (c).

Figure 4.7. Implementation of Algorithm 5 for Multiple Distributions.

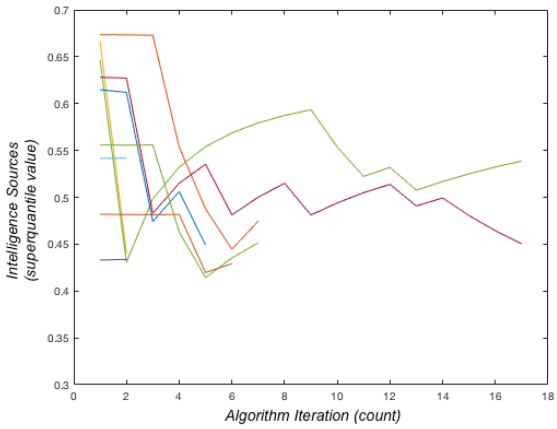
4.2.2 High Sample Trials

In contrast to the previous section, the total number of iterations has not changed for this analysis, however, the number of additional observations in each iteration has. By increasing the number of observations from 10^4 to 10^5 sampled at each iteration, we note that both CVaR elimination algorithms now converge. Due to the stochastic nature of the underlying root finding problem, figure 4.8(b) depicts a convergence in only 18 iterations, as opposed to more than 250 in figure 4.8(c). We do observe that in each case the algorithm stopping criteria is met and promptly exits from any further iterations. This is the expected behavior. Further to this, we increased the number of arms under consideration, depicted in 4.8(a), detailing the convergence of the CVaR elimination algorithm in a mere 160 iterations for the triangular distribution. The number of arms under consideration has been increased to 100: a four-fold increase from our other numerical examples.

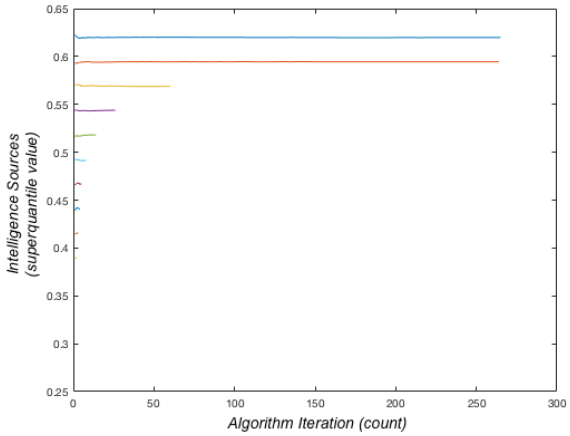
Additionally, we observe that for this replication, a non-optimal arm is selected as optimal as the algorithm stops. This is the 100δ percentage of cases where the optimal arm will not be selected and occurs with a probability of δ. This again illustrates the stochastic nature of the algorithm and the potential for incorrect selection of the *optimal* arm.



$n = 10^5$ per iteration for each iteration.



Triangular distribution, (a) - left and (b) - right.



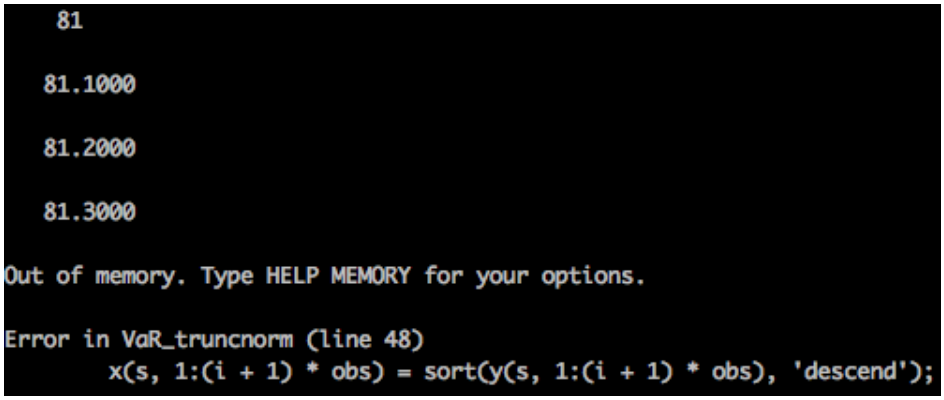
Uniform distribution (c).

Figure 4.8. Implementation of Algorithm 5 for Multiple Distributions.

4.3 High Dimensional Data

Contained in Figure 4.9 is a classic limitation on scalability, here specifically of our implemented superquantile sequential elimination algorithm. As a result of the development of the resulting large and non-sparse matrix, we observe our code gradually utilizing greater resources, until either imposed or physical limits are reached. This is an important consideration for future work,

particularly for implementation on live database systems; see Chapter 5 for a further discussion on this.



Memory error for $n = 10^5$ observations with $1,000$ iterations, occurring at iteration 813 (a matrix of size $81,300,000 \times 100$).

Figure 4.9. Memory Error on the *Hamming* Architecture.

4.4 Algorithm Verification

Contained in Figure 3.1 from the previous chapter we observe the true solution to the stochastic root finding problem posed, shown as the blue line. In Figure 4.10, we have superimposed the empirical estimate for $g(\cdot)$ onto this plot, depicted as the orange line. This depicts the solution of $g(\cdot)$ as a sensitivity analysis for various values of k over the interval $[-100, 100]$. In this instance, $n = 100$ points were evaluated to identify potential errors within our root finding function: the core of our two algorithms. It is clear that even over a large domain such as is presented, the empirical estimate for $g(\cdot)$ is tolerable, within bounds. Of observational note is the conservative nature of the empirical solution, where both the tail decay and algorithm peak do not have the solution range of the true root solution. The total function range is lower in the maximum value and higher in the minimum value for our empirical $g(\cdot)$ when compared to Equation 4.2. Figure 4.10 shows the difference between the true root solution given in Equation 4.2 and our empirical solution for 100 linearly spaced data points.

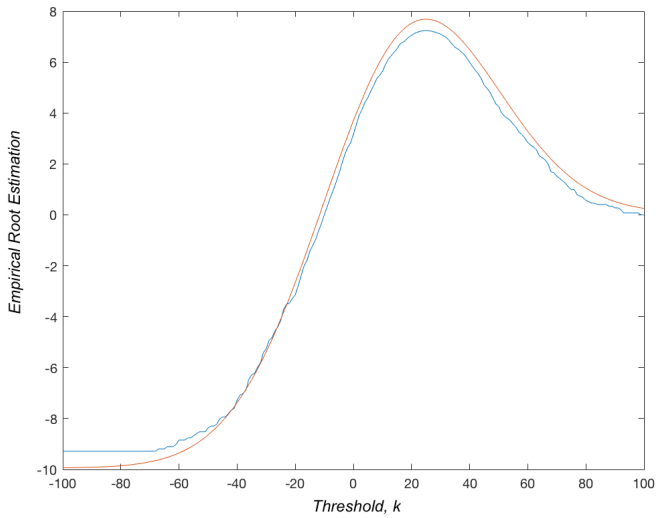
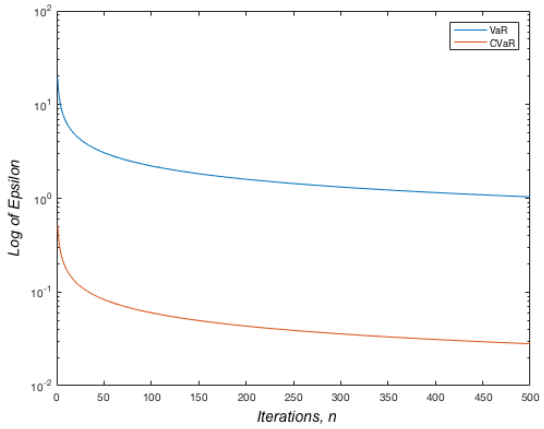


Figure 4.10. Empirical Estimate of $g(\cdot)$ Versus the Root Equation Solution for Values of $k \in -100, \ldots, 100$, Where $n = 100$, $C = 25$, $\mu = 15$, and $\sigma = 30$.

4.5 Convergence of Epsilon

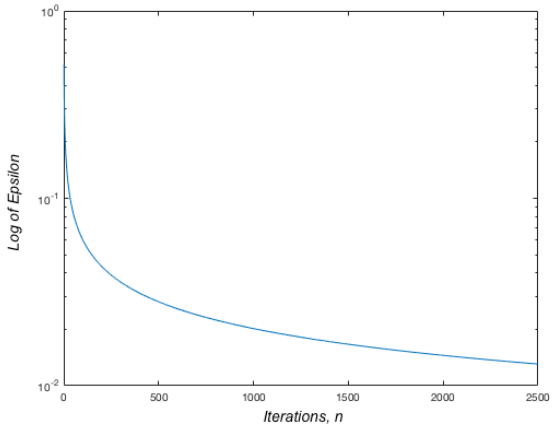
The behavior of the threshold parameter ϵ, as a function of the number of algorithm iterations, is depicted in Figures 4.11 and 4.12. We consider the truncated normal case, for which ϵ is the largest among the three distributions considered, and hence is the worst case. In other words, having ϵ relatively large results in a potentially slower rate of sequential elimination than would otherwise be seen from both the triangular and uniform distributions.

In comparing directly the VaR elimination algorithm and CVaR elimination algorithm, ϵ shown in Figure 4.11, it is observed that both algorithms display the same monotonic decreasing behavior, albeit with different magnitudes at each iteration. The long-run behavior is shown in Figure 4.12, which further illustrates the properties of the parameter ϵ described. As the number of iterations gets large, ϵ approaches 0, ensuring that selection the optimal arm(s) occurs. This has only been executed for the superquantile elimination algorithm and as such, no comparison exists as with Figure 4.11.



Comparison of VaR and CVaR for the parameter ϵ.

Figure 4.11. Numerical Convergence of ϵ.



Numerical depiction of the parameter ϵ over $2,500$ iterations.

Figure 4.12. Long-run Numerical Convergence of ϵ.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Concluding Remarks

In this final chapter, we discuss open issues and future work in relation to the problem presented in this work, both in a critical manner that addresses limitations, as well in the direction forward to support further development in this field to address intelligence processing issues.

We have two main recommendations: first, upon the development of stable and scalable implementations of the algorithms, testing must occur on known data sets with expected outcomes. This is key in verifying and validating outputs on live data, vice the numerical guarantees provided within this thesis. Second, and more importantly from a long-term standpoint, the elimination algorithms should be implemented within an intelligence organization to enhance the analysis capability: this work is a force multiplier.

There exist great opportunities for future work as a result of this thesis, both in the applied and theoretical realms. Two main areas have been identified for future work in the applied domain. The first is extending the algorithm to work on real data sets, such as stock portfolio data. This will verify and validate the theoretical performance on known data, whilst negating the immediate requirement to parallelize the algorithm. The second identified applied area is regarding parallelization and scaling of the algorithm to handle large and non-sparse data sets. This is of critical importance in any real-world application.

Three theoretical opportunities were identified for continuing this work. The first aligns closely with the parallelization advancement discussed above, in which for very large matrices on the order of 10^9 elements, the current storage of every observation is not practical. Work needs to be undertaken to review when it is appropriate to remove observations that are not required and replace them with a tuple of data containing index positions, summary statistics and weightings. This elimination of additional data points will significantly reduce the runtime and storage requirements. The second area for improvement is regarding the proof resulting in the parameters ψ and ζ. While quite conservative in their currently implemented form, we observe that optimization of each parameter is possible for various distributions, as well empirical data. These parameters are critical to improving the runtime of each algorithm. Related to this, work to extend the underlying distributions to an unbounded domain case must occur, as our work so far requires a bounded domain distribution assumption. The removal of this requirement will allow observations from distributions with infinite domains, such as the classic normal distribution.

We have studied a resource allocation problem in an intelligence setting, attempting to enhance

efficiency within the first two stages of the intelligence cycle, and thus improving the quality of the intelligence items that are to be considered by analysts. We created two algorithms to find the source(s) that produce the largest fraction of relevant items with respect to a request for information. More generally, this thesis presented a new approach to identifying the arm(s) with the largest or smallest VaR or CVaR risk, under a loss constraint. This problem is not only important in intelligence applications, but in marketing and finance, as discussed. Some readers may note that definitive conclusions are not presented within this work—this is entirely intentional—as the further work mentioned within this chapter will be required for a critical body of mass to be achieved in this research endeavour. Our contribution has set the conditions for further advancements to be made.

A.1 Proof of Theorem 1

We suppose that $k_n < k(C) - \epsilon$, for $\epsilon > 0$. Given the case $k_n > k(C) + \epsilon$, then, from (3.8),

$$\begin{aligned}
0 &\leq \frac{1}{n}\sum_{i=1}^{n}(X_i - C)\, I\,(X_i > k_n) \\
&= \frac{1}{n}\sum_{i=1}^{n}(X_i - C)\, I\,(k_n < X_i \leq k\,(C)) + \frac{1}{n}\sum_{i=1}^{n}(X_i - C)\, I\,(X_i > k\,(C)) \\
&\leq \frac{1}{n}\sum_{i=1}^{n}(X_i - C)\, I\,(k\,(C) - \epsilon < X_i \leq k\,(C)) + \frac{1}{n}\sum_{i=1}^{n}(X_i - C)\, I\,(X_i > k\,(C))\,,
\end{aligned}$$

as $X_i < C$ on the event $\{X_i < k(C) - \epsilon\}$. It follows that, since $C > k(C)$,

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}(X_i - C)\, I\,(X_i > k\,(C)) \\
&\geq -\frac{1}{n}\sum_{i=1}^{n}(X_i - C)\, I\,(k\,(C) - \epsilon < X_i \leq k\,(C)) \\
&\geq (C - k\,(C))\,\frac{1}{n}\sum_{i=1}^{n} I\,(k\,(C) - \epsilon < X_i \leq k\,(C))\,.
\end{aligned}$$

Then, it must hold that

$$\begin{aligned}
&P\,(k_n < k(C) - \epsilon) \\
&\leq P\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - C)I(X_i > k(C)) \geq (C - k(C))\frac{1}{n}\sum_{i=1}^{n} I(k(C) - \epsilon < X_i \leq k(C))\right) \\
&\leq \exp\left(-2n\left(\frac{(C - k(C))P(k(C) - \epsilon < X_i \leq k(C))}{b - a}\right)^2\right)
\end{aligned}$$

and by Hoeffding's Lemma. Hence, by Assumption A3 and Lemma 1 below,

$$P\,(k_n < k(C) - \epsilon) \leq \exp(-2n\psi^2\epsilon^2\zeta^2/(b - a)^2). \tag{A.1}$$

In proving the other direction, if $k_n > k(C) + \epsilon$ then, Equation 3.8 results in

$$\frac{1}{n}\sum_{i=1}^{n}(X_i - C)I(X_i > k(C) + \epsilon) < 0 \leq \frac{1}{n}\sum_{i=1}^{n}(X_i - C)I(X_i > k_n),$$

where this covers the third possibility for the root k_n discussed in Chapter 3.
Also, since $E[(X - C)I(X > k(C))] = 0$,

$$\begin{aligned} &E[(X - C)I(X > k(C) + \epsilon)] \\ &= E[(C - X)I(k(C) < X \leq k(C) + \epsilon)] \\ &> (C - k(C))P(k(C) < X \leq k(C) + \epsilon) \\ &\geq \psi\epsilon\zeta, \end{aligned} \tag{A.2}$$

by Assumption A3 and Lemma 1. It then follows that

$$\begin{aligned} &P\left(k_n > k(C) + \epsilon\right) \\ &\leq P\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - C)I(X_i > k(C) + \epsilon) < 0\right) \\ &= P\Big(E[(X - C)I(X > k(C) + \epsilon)] - \frac{1}{n}\sum_{i=1}^{n}(X_i - C)I(X_i > k(C) + \epsilon) \geq E[(X - C)I(X > k(C) + \epsilon)]\Big) \\ &\leq P\Big(E[(X - C)I(X > k(C) + \epsilon)] - \frac{1}{n}\sum_{i=1}^{n}(X_i - C)I(X_i > k(C) + \epsilon) \geq \psi\epsilon\zeta\Big) \\ &\leq \exp\left(-2n\left(\frac{\psi\epsilon\zeta}{b - a}\right)^2\right), \end{aligned} \tag{A.3}$$

by (A.2) and Hoeffding's Lemma. In summary, we see that

$$P(|k_n - k(C)| > \epsilon) \leq 2\exp\left(-2n\left(\frac{\psi\epsilon\zeta}{b - a}\right)^2\right).$$

From here, the results are input into the sequential elimination approach of [26], in order to obtain the elimination algorithm, as will be shown. For $0 < \delta < 1$ selected by the agent, set

$$P(|k_n - k(C)| > \epsilon_n) \leq 2\exp\left(-2n\left(\frac{\psi\epsilon\zeta}{b - a}\right)^2\right) = \frac{6\delta}{\pi^2 n^2 S}.$$

Solving for ϵ_n,

$$\epsilon_n = \left(\log\left(\frac{\pi^2 n^2 S}{3\delta}\right)\frac{1}{2n}\right)^{1/2}\frac{b - a}{\zeta\psi}.$$

Thus, for any $n = 1, 2, \ldots$, and ϵ_n as given above,

$$P\left(|k_{s,n} - k_s(C)| > \epsilon_n\right) \leq \frac{6\delta}{\pi^2 n^2 S},$$

so, we therefore obtain that

$$\sum_{n=1}^{\infty} P\left(|k_{s,n} - k_s(C)| > \epsilon_n\right) \leq \frac{\delta}{S} \sum_{n=1}^{\infty} \frac{6}{\pi^2 n^2} = \frac{\delta}{S},$$

and due to Basel's problem,

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

Hence,

$$P\left(\cup_{n,s} |k_{s,n} - k_s(C)| > \epsilon_n\right) \leq \sum_{s,n} P\left(|k_{s,n} - k_s(C)| > \epsilon_n\right) \leq \sum_{s=1}^{S} \frac{\delta}{S} \leq \delta.$$

It follows that,

$$P\left(|k_{s,n} - k_s(C)| \leq \epsilon_n, \forall n, \forall s = 1, \ldots, S\right) \geq 1 - \delta.$$

Lemma 1 *Setting ψ to*

$$\psi = \frac{\frac{b-C}{2} \frac{b-C}{2} \zeta}{1 - \frac{b-C}{2} \zeta} > 0$$

satisfies $C - k(C) \geq \psi$.

Proof of Lemma 1: We argue that

$$E[X \mid X > C - \psi] \geq C$$

which implies that $C - k(C) \geq \psi$. Indeed,

$$\begin{aligned}
E[X \mid X > C - \psi] &= \frac{E[XI(X > C - \psi)]}{P[X > C - \psi]} \\
&= \frac{E[XI(C - \psi < X \leq C + \frac{b-C}{2})]}{P[X > C - \psi]} + \frac{E[XI(X > C + \frac{b-C}{2})]}{P[X > C - \psi]} \\
&\geq \frac{E[(C - \psi)I(C - \psi < X \leq C + \frac{b-C}{2})]}{P[X > C - \psi]} + \frac{E[(C + \frac{b-C}{2})I(X > C + \frac{b-C}{2})]}{P[X > C - \psi]} \\
&= (C - \psi)\frac{P[C - \psi < X \leq C + \frac{b-C}{2}]}{P[X > C - \psi]} + (C + \frac{b - C}{2})\frac{P[X > C + \frac{b-C}{2}]}{P[X > C - \psi]} \\
&= (C - \psi)\left(1 - \frac{P[X > C + \frac{b-C}{2}]}{P[X > C - \psi]}\right) + (C + \frac{b - C}{2})\frac{P[X > C + \frac{b-C}{2}]}{P[X > C - \psi]} \\
&\geq (C - \psi)\left(1 - P[X > C + \frac{b - C}{2}]\right) + (C + \frac{b - C}{2})P[X > C + \frac{b - C}{2}] \\
&\geq (C - \psi)\left(1 - P[X > C + \frac{b - C}{2}]\right) + (C + \frac{b - C}{2})(b - C - \frac{b - C}{2})\zeta \\
&\geq (C - \psi)\left(1 - \frac{b - C}{2}\zeta\right) + (C + \frac{b - C}{2})(\frac{b - C}{2})\zeta \\
&\geq C - \psi\left(1 - \frac{b - C}{2}\zeta\right) + \frac{b - C}{2}\frac{b - C}{2}\zeta.
\end{aligned}$$

We must ensure that ψ is small enough so that the right hand side is at least C. By inspection,

$$\psi \leq \frac{\frac{b-C}{2}\frac{b-C}{2}\zeta}{1 - \frac{b-C}{2}\zeta},$$

which completes the proof.

A.2 Proof of Theorem 2

Equation 3.16 leads to,

$$P\Big(\alpha_n - \alpha > \epsilon\Big) \leq P\Big(\bar{F}(k_n) - \bar{F}(k(C)) > q\epsilon\Big) + P\Big(\bar{F}(k(C)) - F(k(C)) > (1-q)\epsilon\Big), \quad \text{(A.4)}$$

for $0 < q < 1$ and $\epsilon > 0$. For the first term

$$P(\bar{F}(k_n) - \bar{F}(k(C)) > q\epsilon) = P\left(\frac{1}{n}\sum_{i=1}^{n} I(k(C) < X_i \leq k_n) > q\epsilon\right).$$

If $(1/n)\sum_{i=1}^{n} I(k(C) < X_i \leq k_n) > q\epsilon$ then

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}(X_i - C)I(X_i > k_n) - \frac{1}{n}\sum_{i=1}^{n}(X_i - C)I(X_i > k(C)) \\
&= \frac{1}{n}\sum_{i=1}^{n}(C - X_i)I(k(C) < X_i \leq k_n) \\
&\geq (C - k_n)q\epsilon.
\end{aligned}$$

Since $0 \leq (1/n)\sum_{i=1}^{n}(X_i - C)I(X_i > k_n) \leq (b-C)/n$, for $0 < \xi < \psi$,

$$\begin{aligned}
&P\left(\frac{1}{n}\sum_{i=1}^{n} I(k(C) < X_i \leq k_n) > q\epsilon\right) \\
&\leq P\left(\frac{1}{n}\sum_{i=1}^{n}(C - X_i)I(X_i > k(C)) > (C - k_n)q\epsilon - (b-C)/n\right) \\
&= P\left(\frac{1}{n}\sum_{i=1}^{n}(C - X_i)I(X_i > k(C)) > (C - k_n)q\epsilon - (b-C)/n; k_n - k(C) > \xi\right) \\
&\quad + P\left(\frac{1}{n}\sum_{i=1}^{n}(C - X_i)I(X_i > k(C)\right) > (C - k_n)q\epsilon - (b-C)/n; k_n - k(C) \leq \xi) \\
&\leq P\left(k_n - k(C) > \xi\right) + P\left(\frac{1}{n}\sum_{i=1}^{n}(C - X_i)I(X_i > k(C)) > (\psi - \xi)q\epsilon - (b-C)/n\right),
\end{aligned}$$

by Assumption A3 and Lemma 1. Hence,

$$P(\bar{F}(k_n) - \bar{F}(k(C)) > q\epsilon) \leq \exp\left(-2n\left(\frac{\psi\xi\zeta}{b-a}\right)^2\right) + \exp\left(-2n\left(\frac{(\psi-\xi)q\epsilon - (b-C)/n}{b-a}\right)^2\right),$$

and again by Equation A.3 and Hoeffding's Lemma, for $n > (b-C)/((\psi-\xi)q\epsilon)$. Subsequently, regarding the second term in Equation A.4,

$$P(\bar{F}(k(C)) - F(k(C)) > (1-q)\epsilon) \leq \exp(-2n(1-q)^2\epsilon^2).$$

Thus, in summary we observe that

$$\begin{aligned}
&P(\alpha_n - \alpha > \epsilon) \\
&\leq \exp\left(-2n\left(\frac{\psi\xi\zeta}{b-a}\right)^2\right) + \exp\left(-2n\left(\frac{(\psi-\xi)q\epsilon - (b-C)/n}{b-a}\right)^2\right) + \exp(-2n(1-q)^2\epsilon^2) \\
&\leq 3\exp\left(-2n\left(\min\left\{\frac{\psi\xi\zeta}{b-a}, \frac{(\psi-\xi)q\epsilon - (b-C)/n}{b-a}, (1-q)\epsilon\right\}\right)^2\right).
\end{aligned}$$

Whilst unoptimised in this work $\xi = \psi/2$ and $q = 1/2$ (note: an optimisation of these parameters could occur in future work), so that

$$P(\alpha_n - \alpha > \epsilon) \leq 3\exp\left(-2n\left(\min\left\{\frac{\psi^2\epsilon\zeta}{2(b-a)}, \frac{\psi\epsilon/2 - (b-C)/n}{2(b-a)}, \epsilon/2\right\}\right)^2\right),$$

for $n > 2(b-C)/(\psi\epsilon)$. The analysis of $P(\alpha_n < \alpha - \epsilon)$ is similar and results in an identical exponential bound; the proof is omitted for the sake of brevity. The conclusion we obtain is that,

$$P(|\alpha_n - \alpha| > \epsilon) \leq 6\exp\left(-2n\left(\min\left\{\frac{\psi^2\epsilon\zeta}{2(b-a)}, \frac{\psi\epsilon/2 - (b-C)/n}{2(b-a)}, \epsilon/2\right\}\right)^2\right),$$

for $n > 2(b-C)/(\psi\epsilon)$. As in the proof of Theorem 1, for $0 < \delta < 1$ chosen by the agent, ϵ_n is set so that

$$6\exp\left(-2n\left(\min\left\{\frac{\psi^2\epsilon_n\zeta}{2(b-a)}, \frac{\psi\epsilon_n/2 - (b-C)/n}{2(b-a)}, \epsilon_n/2\right\}\right)^2\right) \leq \frac{6\delta}{\pi^2 n^2 S},$$

which leads to,

$$\epsilon_n = \left(\log\left(\frac{\pi^2 n^2 S}{\delta}\right)\frac{2}{n}\right)^{1/2} \max\left\{\frac{b-a}{\psi^2\zeta}, \frac{2(b-a) + (b-C)/n)}{\psi}, 1\right\}.$$

By standard arguments, as in the proof of Theorem 1, it follows that,

$$P\left(|\alpha_{s,n} - \alpha_s(C)| \leq \epsilon_n, \forall n, \forall s = 1, \ldots, S\right) \geq 1 - \delta.$$

A.3 MATLAB Implementation of Algorithm 4

```
function [result, epsilon, verbose_arms, mu, optimal_arm, root_max,
    expected_samples]...
    = VaR_truncnorm(C, sigma, a, b, S, delta, obs, max_iter)
%
% Adam J Hepworth
% Naval Postgraduate School
%
verbose_arms = true(1, S);
mu = linspace(a + 1, C - 1, S);
y = zeros(size(mu, 2),(max_iter + 1) * obs);
x = zeros(size(mu, 2), (max_iter + 1) * obs);
result = zeros(size(mu, 2), max_iter + 1);
epsilon = zeros(1, max_iter + 1);
i = 0;
zeta = min(min(normpdf((a - mu)/sigma), normpdf((b - mu)/sigma))...
    ./ (sigma * (normcdf((b - mu)/sigma) - normcdf((a - mu)/sigma))));
for s = 1:size(mu,2)
    root_true(s) = fzero(@(k) sigma * (normpdf((k - mu(s))/sigma)...
        - normpdf((b - mu(s))/sigma)) + mu(s) .* (normcdf((b - mu(s))/sigma)
    ...
        - normcdf((k - mu(s))/sigma)) - C * (normcdf((b - mu(s))/sigma)...
        - normcdf((k - mu(s))/sigma)), [a, C]);
end
psi = min(C - root_true);
while(sum(double(verbose_arms)) > 1) && (i < max_iter)
    for s = 1:size(mu, 2)
        y(s, i * obs + (1:obs)) = random(truncate(makedist...
            ('Normal', mu(s), sigma), a, b), [1, obs]);
        x(s, 1:(i + 1) * obs) = sort(y(s, 1:(i + 1) * obs), 'descend');
        root_eval = size(find(cumsum(x(s, 1:(i + 1) * obs) - C) > 0), 2);
        if (root_eval > 0)
            result(s, i + 1) = x(s, root_eval);
        else
            result(s, i + 1) = a;
        end
    end
    epsilon(1 + i) = sqrt((.5/(obs * (i + 1)))...
        * log((pi^2 * (obs * (i + 1))^2 * S)/(3 * delta)))...
        * (b - a)/(zeta * psi);

    root_max(i + 1) = max(result(:, i + 1));
    optimal_arm(i + 1) = find(result(:, i + 1) == root_max(i + 1));
    for s = 1:size(mu, 2)
```

```
        if (verbose_arms(s) == true) && (double(size(verbose_arms(1:s), 2))
    ~= optimal_arm(i + 1))
            if abs(root_max(i + 1) - result(s, i + 1)) > (2 * epsilon(1 + i))
                result(s, i + 1) =  NaN;
                verbose_arms(s)  = false;
            end
        elseif(verbose_arms(s) == false)
            result(s, i + 1) = NaN;
        end
    end
    expected_samples(i + 1) = ((8*(b - a)^2) / (psi^2 * zeta^2))...
        * log((pi^2 * S) / (3 * delta))...
        * ((double(size(verbose_arms(1:s), 2) - 1)...
        * (4 * epsilon(i + 1))^(-2)));
    disp(i/max_iter*100)
    i = i + 1;
end
save('VaR_truncnorm_data.mat');
end
%
% end of program
%
```

Listing 1: Implementation of the Sequential Quantile Elimination Algorithm for the Truncated Normal Distribution

A.4 MATLAB Implementation of Algorithm 5

```
function [result, epsilon, verbose_arms, mu, optimal_arm, root_max,
    expected_samples]...
    = CVaR_truncnorm(C, sigma, a, b, S, delta, obs, max_iter)
%
% Adam J Hepworth
% Naval Postgraduate School
%
verbose_arms = true(1, S);
mu = linspace(a + 1, C - 1, S);
y = zeros(size(mu, 2),(max_iter + 1) * obs);
x = zeros(size(mu, 2), (max_iter + 1) * obs);
result = zeros(size(mu, 2), max_iter + 1);
epsilon = zeros(1, max_iter + 1);
i = 0;
zeta = min(min(normpdf((a - mu)/sigma), normpdf((b - mu)/sigma))...
    ./ (sigma * (normcdf((b - mu)/sigma) - normcdf((a - mu)/sigma))));
for s = 1:size(mu,2)
    root_true(s) = fzero(@(k) sigma * (normpdf((k - mu(s))/sigma)...
        - normpdf((b - mu(s))/sigma)) + mu(s) .* (normcdf((b - mu(s))/sigma)
    ...
        - normcdf((k - mu(s))/sigma)) - C * (normcdf((b - mu(s))/sigma)...
        - normcdf((k - mu(s))/sigma)), [a, C]);
end
psi = min(C - root_true);
while(sum(double(verbose_arms)) > 1) && (i < max_iter)
    for s = 1:size(mu, 2)
        y(s, i * obs + (1:obs)) = random(truncate(makedist...
            ('Normal', mu(s), sigma), a, b), [1, obs]);
        x(s, 1:(i + 1) * obs) = sort(y(s, 1:(i + 1) * obs), 'descend');
        root_eval = size(find(cumsum(x(s, 1:(i + 1) * obs) - C) > 0), 2);
        if (root_eval > 0)
            result(s, i + 1) = (((i + 1) * obs) - root_eval) / ((obs * (i + 1)
    ));
        else
            result(s, i + 1) = 0;
        end
    end
    epsilon(1 + i) = real(sqrt((2 / (obs * (i + 1)))...
        * log((pi^2 * (obs * (i + 1))^2 * S)/delta))...
        * max([((b - a)/(zeta * psi^2)); ((2 * (b - a) + (b - C)...
        /(obs * (i + 1)))/psi); 1]));
    root_max(i + 1) = max(result(:, i + 1));
    optimal_arm(i + 1) = find(result(:, i + 1) == root_max(i + 1));
```

```
    for s = 1:size(mu, 2)
        if (verbose_arms(s) == true) && (double(size(verbose_arms(1:s), 2)) ~=
    optimal_arm(i + 1))
            if abs(root_max(i + 1) - result(s, i + 1)) > (2 * epsilon(1 + i))
                result(s, i + 1) =  NaN;
                verbose_arms(s)  = false;
            end
        elseif(verbose_arms(s) == false)
            result(s, i + 1) = NaN;
        end
    end
    expected_samples(i + 1) = 32 * (max([((b - a)/(zeta * psi^2));...
        ((2 * (b - a) + (b - C)/(obs * (i + 1)))/psi); 1]))^2 ...
        * log((pi^2 * S) / (3 * delta))...
        * ((double(size(verbose_arms(1:s), 2) - 1)...
        * (4 * epsilon(i + 1))^(-2)));
    disp(i/max_iter*100)
    i = i + 1;
end
save('CVaR_truncnorm_data.mat');
end
%
% end of program
%
```

Listing 2: Implementation of the Sequential Superquantile Elimination Algorithm for the Truncated Normal Distribution

List of References

[1] *Joint Intelligence*, JP 2-0, U.S. Joint Chiefs of Staff, Washington, DC, 2013.

[2] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, pp. 201–227, 1999.

[3] R. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking and Finance*, vol. 26, pp. 1443–1471, 2002.

[4] A. Ruszczynski and A. Shapiro, "Optimization of convex risk functions," *Mathematics of Operations Research*, vol. 31(3), pp. 433–452, 2006.

[5] R. Rockafellar and S. Uryasev, "The fundamental risk quadrangle in risk management, optimization and statistical estimation," *Surveys in Operations Research and Management Science*, vol. 18, pp. 33–53, 2013.

[6] S. Uryasev, "Var and cvar in risk management and optimization," *European International CARISMA Conference*, September 2010, presentation unpublished.

[7] R. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, pp. 21–42, 2000.

[8] D. Brown, "Large deviations bounds for estimating conditional value-at-risk," *Operations Research Letters*, vol. 35, pp. 722–730, 2007.

[9] R. Pasupathy and S. Kim, "The stochastic root-finding problem: Overview, solutions, and open questions," *ACM Transactions on Modeling and Computer Simulation*, vol. 21, no. 3, 2011.

[10] B. Szörényi, R. Busa-Fekete, P. Weng, and E. Hüllermeier, "Qualitative multi-armed bandits: A quantile-based approach," *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015.

[11] Y. David and N. Shimkin, "Pure exploration for max-quantile bandits," *Machine Learning and Knowledge Discover in Databases, European Conference, EMCL PKDD Proceedings*, pp. 556–571, 2016.

[12] Mathworks, *Matlab*. Natick, MA: The MathWorks, Inc., 2017.

[13] S. Shalev-Shwartz, "Online learning: Theory, algorithms, and applications," Ph.D. dissertation, Hebrew University, Jerusalem, Israel, July 2007.

[14] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[15] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, Massachusetts: The MIT Press, 2016.

[16] T. Lai and H. Robbins, "Asymptotically efficient allocation rules," *Advances in Applied Mathematics*, vol. 6, pp. 4 – 22, 1985.

[17] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and non-stochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, 2012.

[18] E. Even-Dar, S. Mannor, and Y. Mansour, "PAC bounds for multi-armed bandit and markov decision processes," *In Fifteenth Annual Conference on Computational Learning Theory (COLT)*, pp. 255–270, 2002.

[19] S. Mannor and J. Tsitsiklis, "The sample complexity of exploration in the multi-armed bandit problem," *Journal of Machine Learning Research*, vol. 5, pp. 623–648, 2004.

[20] P. Glynn and S. Juneja, "Ordinal optimization - empirical large deviations rate estimators, and stochastic multi-armed bandits," *arXiv:1507.04564*, 2015.

[21] R. T. Rockafellar and J. O. Royset, "Engineering decisions under risk-averseness," *Proceedings of International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP)*, 2015.

[22] S. Uryasev, "Buffered probability of exceedance and buffered service level: Definitions and properties," 2014, research report.

[23] R. Rockafellar and J. Royset, "On buffered failure probability in design and optimization of structures," *Reliability Engineering and System Safety*, vol. 95, pp. 499–510, 2010.

[24] R. Serfling, *Approximation Theorems of Mathematical Statistics*. New York City, NY: John Wiley & Sons, 2008.

[25] T. Cormen, C. Leiserson, R. Rivset, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, Massachusetts: MIT Press, 2010.

[26] E. Even-Dar, S. Mannor, and Y. Mansour, "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems," *Journal of Machine Learning Research*, vol. 7, pp. 1079 – 1105, 2006.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California